

Technical Report 1702  
November 1995

## *SeaRad*, A Sea Radiance Prediction Code

C. R. Zeisse

Naval Command, Control and  
Ocean Surveillance Center  
RDT&E Division

San Diego, CA  
92152-5001

19960129 119

DTIC QUALITY INSPECTED 1



Approved for public release; distribution is unlimited.



**Technical Report 1702**  
November 1995

***SeaRad*, A Sea Radiance  
Prediction Code**

C. R. Zeisse



**NAVAL COMMAND, CONTROL AND  
OCEAN SURVEILLANCE CENTER  
RDT&E DIVISION  
San Diego, California 92152-5001**

---

**K. E. EVANS, CAPT, USN**  
**Commanding Officer**

**R. T. SHEARER**  
**Executive Director**

**ADMINISTRATIVE INFORMATION**

The work detailed in this report was performed for the Office of Naval Research by the Naval Command, Control and Ocean Surveillance Center, RDT&E Division, Tropospheric Branch, Code 543. Funding was provided under Program Element 0602435N, Accession Number DN302215.

Released by  
R. A. Paulus, Head  
Tropospheric Branch

Under authority of  
J. H. Richter, Head  
Ocean and Atmospheric Sciences  
Division



## EXECUTIVE SUMMARY

### OBJECTIVE

Develop a computer code to predict sea radiance (brightness).

### APPROACH

Sea radiance is modeled by combining the methods of geometrical optics with the Cox-Munk statistical description of ocean capillary waves. The model is incorporated into the atmospheric transmittance/radiance code MODTRAN2 to provide numerical sea radiance predictions.

In this model each individual capillary wave facet is allowed to reflect the sky or sun and emit thermal radiation. The total radiance from the sea is obtained by applying the proper statistical weight to each facet and integrating over all facets within the observer's field-of-view.

### RESULTS

The modified MODTRAN2 code, called *SeaRad*, calculates sea radiance for any viewing geometry in a spectral range from  $52.63 \text{ cm}^{-1}$  to  $25000 \text{ cm}^{-1}$ . Typical execution speeds are approximately 10 s per pixel on a Pentium/90 MHz personal computer. Preliminary comparisons show that *SeaRad* agrees to within several degrees Celsius ( $^{\circ}\text{C}$ ) with actual sea radiance measurements in the mid-wave and long-wave infrared bands.



## CONTENTS

EXECUTIVE SUMMARY .....	i
1. INTRODUCTION .....	1
2. HARDWARE CONSIDERATIONS .....	1
3. AN EXAMPLE .....	1
4. THE MODEL .....	2
5. THE COORDINATE SYSTEM .....	5
6. SPECULAR REFLECTION .....	6
7. THE OCCURRENCE PROBABILITY .....	6
8. THE INTERACTION PROBABILITY .....	7
9. EQUATIONS FOR SEA RADIANCE .....	8
10. <i>SEARAD</i> .....	11
11. CONCLUSION .....	15
12. REFERENCES .....	16
APPENDIX A: <i>SeaRad</i> INPUT AND OUTPUT FILES .....	A-1
APPENDIX B: MODIFIED SUBROUTINE "DRIVER" .....	B-1
APPENDIX C: MODIFIED SUBROUTINE "TRANS" .....	C-1
APPENDIX D: SOURCE CODE FOR NEW <i>SeaRad</i> SUBROUTINES .....	D-1

## Figures

1. Four contributions to sea radiance .....	3
2. Coordinate system for facet reflection .....	5
3. Cox-Munk occurrence probability density for a windspeed of $10 \text{ m s}^{-1}$ .....	7
4. Cox-Munk-Plass interaction probability density for a windspeed of $10 \text{ m s}^{-1}$ .....	8
5. Flowchart for modifications to MODTRAN2 subroutine "DRIVER" .....	12
6. Flowchart for modifications to MODTRAN2 subroutine "DPFNMN" .....	13
7. Flowchart for <i>SeaRad</i> subroutine "Card" .....	14
B-1. Detailed flowchart for modified subroutine "DRIVER" .....	B-2
C-1. Flowchart for modified subroutine "TRANS" .....	C-2



## 1. INTRODUCTION

*SeaRad* is a FORTRAN computer code that predicts the radiance (brightness) of the ocean surface. *SeaRad* is based on the Cox-Munk statistical model (Cox and Munk, 1954, 1956) for wind-driven capillary wave facets. An individual facet is chosen and assigned a specific slope with respect to the local horizon. The facet is allowed to reflect the sky and sun and emit thermal black body radiation toward an observer. The total radiance is obtained by applying the proper statistical weight to the facet and integrating over all facets within the observer's field-of-view.

*SeaRad* is valid for a spectral range extending from the visible to the far infrared. Preliminary comparisons show that *SeaRad* agrees to within several °C with actual sea radiance measurements in the mid-wave and long-wave infrared bands.

In its current form, *SeaRad* is a self-contained, DOS-compatible program that runs on a personal computer and computes radiance for a single pixel (rather than an entire image). It is a modified version of the Air Force program MODTRAN2 (Berk, et al., 1989; Kneizys, et al., 1988) that computes atmospheric transmittance and radiance. *SeaRad* operates exactly like the original MODTRAN2 code<sup>1</sup> except that a new logical parameter, "SeaSwitch", is required in the input file. Sun glint is included in the sea radiance prediction provided that the user has chosen to execute *SeaRad* in radiance mode with solar scattered radiance included (IEMSCT = 2).

## 2. HARDWARE CONSIDERATIONS

The size of the FORTRAN source code is 1.8 MB. When assembled by version 5.01 of the Lahey F77L/EM-32 DOS compiler, the size of the executable code is 0.8 MB. When run<sup>2</sup> on a Pentium/90 at low spectral resolution (LOWTRAN7) in multiple scattering mode, execution times are 4 s for a typical thermal long-wave case (830 to 1250 cm<sup>-1</sup> in 21 spectral steps) and 17 s for a typical solar mid-wave case (2000 to 3340 cm<sup>-1</sup> in 67 spectral steps). Source and executable codes are available on disk through correspondence with the author.

## 3. AN EXAMPLE

This section provides an example of how *SeaRad* is used to predict radiance of the ocean surface. An input file called "Tape5Rad.Std," shown on page A-2, employs a 1976 U. S. standard atmosphere to calculate ocean radiance observed at a zenith angle of 100° (a depression angle of 10°) from a height of 23 m. The Navy aerosol model is used. The calculation is done at low spectral resolution (LOWTRAN7) for a single wave number (945 cm<sup>-1</sup>) in the long-wave band.

With this file present, the following three DOS commands will calculate ocean radiance and display results:

```
Copy Tape5Rad.Std Tape5
SeaRad
Type Out
```

- 
1. This report assumes that the reader is familiar with MODTRAN2 operation.
  2. The compiler requires the Lahey/Phar Lap 386 DOS Extender program (0.2 MB) to run on a personal computer.



These commands<sup>3</sup> produce the output file "Out" (Appendix A, page 3). Band-integrated radiance values in  $\text{W m}^{-2} \text{sr}^{-1}$  are listed at the end of the output file for each of four contributions to ocean radiance: path to footprint, sea emission, sky reflection, and sun glint. (In fact, no sun glint has been calculated in this instance since the input file specifies  $\text{IEMSCT} = 1$  rather than  $\text{IEMSCT} = 2$ .) Please note that the parameter "TBOUND" in the input file has been reinterpreted by *SeaRad* as the sea temperature.

The input file shown in Appendix A on page 2 contains two new parameters at the end of the third line: "90.000" and "T". These will be discussed in reverse order of their appearance.

The "T", which may appear anywhere in columns 76 through 80 of the third line of the input file (at the end of Card 3), is a new logical parameter "SeaSwitch". It is required; that is, a fatal error will be generated if it is not present in the input file. "SeaSwitch" controls the sea radiance calculation. When "SeaSwitch" is equal to "T", the sea radiance calculation will be allowed provided certain other conditions are met. When "SeaSwitch" is equal to "F", the sea radiance calculation will be prevented under all conditions and the program will execute as originally released by the Air Force.

The "90.000", which may appear anywhere in columns 66 through 75 of the third line of the input file (near the end of Card 3), is a new floating point parameter, "Psi". It is optional; that is, the program will run whether this parameter is included in the input file or not. "Psi" is the azimuth of the upwind direction<sup>4</sup> measured from the line-of-sight in degrees positive East of North. If it is omitted (if the field is blank), and if all conditions for a sea radiance calculation are met, that calculation will proceed under the assumption that the value of "Psi" is zero, meaning that the observer is looking directly into the wind. For the input file in Appendix A, "Psi" is 90°, meaning that the wind is blowing from right to left, perpendicular to the direction of observation.

The modified version of Card 3 used by *SeaRad* is:

H1, H2, ANGLE, RANGE, BETA, R0, LEN, Psi, SeaSwitch  
Format (6F10.3, I5, F10.3, L5)

## 4. THE MODEL

The primary assumption of the model is that the strength of interaction between an optical ray and a capillary wave facet is given by the facet area projected normal to the ray. A feature (Zeisse, 1994, 1995) of the equations contained in *SeaRad* is that they predict a finite horizon radiance. *SeaRad* does not include multiple reflections, shadowing, or gravity waves. Polarization is ignored.

The model computes four contributions to sea radiance. Each of the four contributions is shown in figure 1. (For purposes of clarity, only two dimensions have been used in figure 1; however, all three dimensions are used in the actual calculation.)

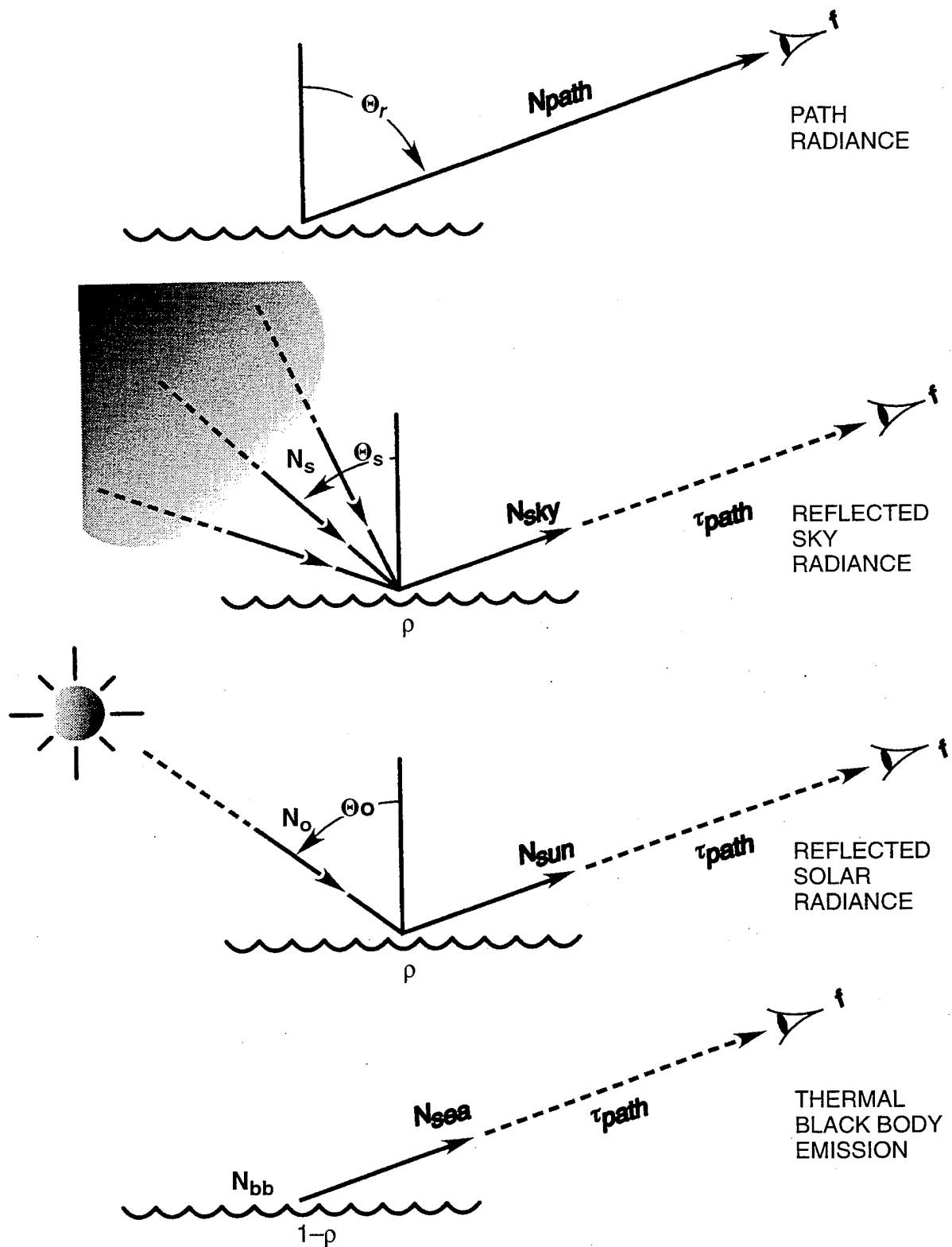
The first contribution is path radiance, shown at the top of figure 1. The footprint of a single pixel in an image of the sea is indicated by the wavy line. The footprint is observed by a receiver at the end

---

3. The time for this particular test case was 3 s on a 486/50 MHz personal computer.

4. This information is required because the Cox-Munk capillary wave statistics are different in the upwind and crosswind directions.





$$(N_{sky} + N_{sun} + N_{sea}) \tau_{path} \cdot f + N_{path} \cdot f = N$$

Figure 1. Four contributions to sea radiance.



of a ray whose zenith angle at the footprint is  $\theta_r$ . Let  $N_{path}$  designate the spectral radiance in  $\text{W m}^{-2} \text{sr}^{-1} (\text{cm}^{-1})^{-1}$  along the path<sup>5</sup> from the footprint to the receiver.

The second contribution is reflected sky radiance. Spectral radiance  $N_s$  from a portion of the sky arrives at the footprint along a ray whose zenith angle there is  $\theta_s$ . The footprint contains wave facets of different slopes, many that reflect the incoming sky radiance away from the receiver toward other parts of the sky. These facets are ignored. However, the footprint will contain some facets whose slope is correct for reflecting the incoming sky radiance toward the receiver along the path defined by the zenith angle  $\theta_r$ . These facets are retained. The contributions from all portions of the sky are summed together after specular reflection by the appropriate facets within the footprint, and the sum leaving the footprint at zenith angle  $\theta_r$  is designated  $N_{sky}$ . During its path to the receiver, the reflected sky radiance is attenuated by the path transmission  $\tau_{path}$ .

The third contribution is reflected solar radiance, sun glint. The calculation is analogous to the calculation of sky radiance. Spectral radiance  $N_o$  from the solar center arrives at the footprint along a path whose zenith angle there is  $\theta_o$ . Within the footprint, most facets deflect the solar ray away from the receiver and are rejected, but some facets are retained because they deflect the ray specularly toward the receiver along a path with zenith angle  $\theta_r$ .  $N_{sun}$  is the spectral radiance leaving the footprint after summation over rays arriving from all portions of the solar disk. The reflected solar radiance is also attenuated by the path transmission  $\tau_{path}$  before final reception.

The fourth contribution is thermal black body emission. Each facet emits a spectral radiance  $N_{bb}$  given by Planck's equation for a black body whose temperature is equal to the value of "TBOUND" in the input file. The spectral emissivity of a given facet in the direction of the receiver is specified by the slope of that facet and the value of  $\theta_r$ .  $N_{sea}$  is the thermal spectral radiance leaving the footprint for the receiver after summation over all facets within the footprint. As before,  $N_{sea}$  is attenuated by path transmission after leaving the footprint.

Throughout figure 1, the symbol  $\rho$  represents the spectral reflectivity of sea water, which is required for the second and third contributions since they are governed by the process of optical reflection. On the other hand, the fourth contribution is governed by the process of optical emission. Fortunately, by application of Kirchhoff's Law to an opaque medium, sea water, the emissivity is given by one minus the reflectivity. The reflectivity is calculated from Fresnel's equations (Stratton, 1941) with a complex optical index taken from the literature (Hale & Querry, 1973; Querry, et al., 1977). These data for the index, available between  $52.63 \text{ cm}^{-1}$  and  $25000 \text{ cm}^{-1}$ , set the spectral range of *SeaRad*.

The total spectral radiance  $N$  received at wave number  $\nu$  ( $\text{cm}^{-1}$ ) is given by

$$N(\nu) = N_{path}(\nu) f(\nu) + \left[ N_{sky}(\nu) + N_{sun}(\nu) + N_{sea}(\nu) \right] \tau_{path}(\nu) f(\nu), \quad (1)$$

where  $f(\nu)$  has been introduced to represent the spectral responsivity of the receiver.

The design of *SeaRad* is such that path ( $N_{path}$ ,  $\tau_{path}$ ) and source ( $N_s$ ,  $N_o$ ,  $N_{bb}$ ) values are taken from the original MODTRAN2 while Fresnel reflection ( $\rho$ ) and slope integrated values ( $N_{sky}$ ,  $N_{sun}$ ,  $N_{sea}$ ) are introduced in new subroutines. Integration of (1) over the wave number band specified in the input file is carried out in a modification of subroutine "TRANS" to produce the band-integrated values for sea radiance given in the output file.

---

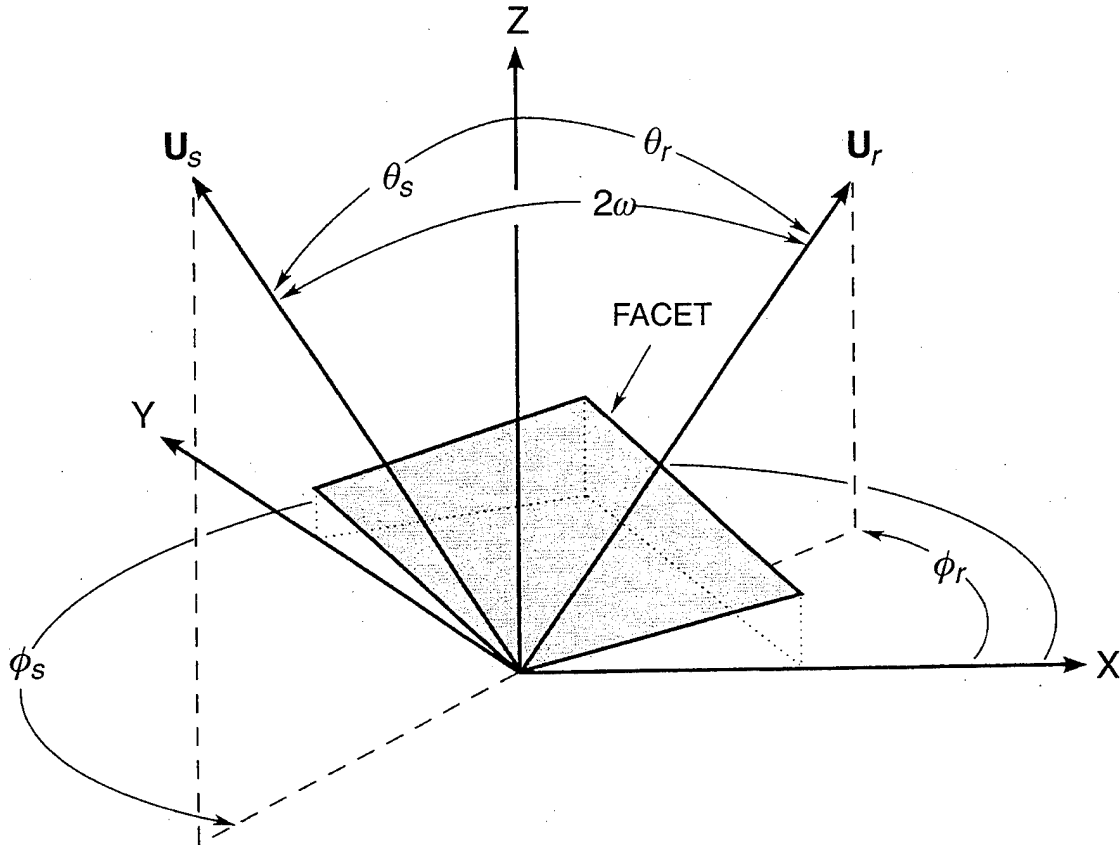
5. In this report, the word path refers to only the optical path between the footprint and the receiver.



## 5. THE COORDINATE SYSTEM

The previous description neglected the azimuthal dependence of rays arriving and leaving the footprint. The full three-dimensional geometry will now be introduced.

Figure 2 shows the geometry of reflection. A coordinate system was chosen whose origin is the point of reflection with the X-axis pointing upwind, the Z-axis pointing toward the zenith, and the Y-axis pointing crosswind such that a right-handed system is formed. The X-Y plane is horizontal at the point of reflection. The tilted facet passes through the origin.



**Figure 2.** Coordinate system for facet reflection.

Define a unit vector  $\mathbf{U} \equiv (\theta, \phi)$  with polar coordinates  $\theta$ , the zenith angle, and  $\phi$ , the azimuth. If we denote the Cartesian coordinates of  $\mathbf{U}$  by  $(a, b, c)$ , then we have

$$\begin{aligned} a &= \sin \theta \cos \phi \\ b &= \sin \theta \sin \phi \\ c &= \cos \theta \end{aligned} \tag{2}$$

for the Cartesian coordinates of  $\mathbf{U}$  in terms of its spherical coordinates and

$$\begin{aligned} \theta &= \cos^{-1}(c) \\ \phi &= \tan^{-1}(b/a) \end{aligned} \tag{3}$$



for the spherical coordinates of  $\mathbf{U}$  in terms of its Cartesian coordinates. Two unit vectors are shown in figure 2:  $\mathbf{U}_s$ , pointing from the origin to the source, and  $\mathbf{U}_r$ , pointing from the origin to the receiver. A third unit vector,  $\mathbf{U}_n$ , is normal to the facet at the point of reflection but was removed from the figure for clarity<sup>6</sup>.

The facet slope in the upwind direction,  $\xi_x$ , is given by the slope of the line formed at the intersection of the facet with the  $X$ - $Z$  plane. The facet slope in the crosswind direction,  $\xi_y$ , is given by the slope of the line formed at the intersection of the facet with the  $Y$ - $Z$  plane. In terms of the Cartesian coordinates of the facet normal these slopes are

$$\begin{aligned}\xi_x &= -a_n/c_n \\ \xi_y &= -b_n/c_n\end{aligned}\tag{4}$$

## 6. SPECULAR REFLECTION

If a specular reflection occurs, the three vectors for source, receiver, and facet normal are connected by the law of reflection:

$$\mathbf{U}_s + \mathbf{U}_r = 2 \cos \omega \mathbf{U}_n\tag{5}$$

where  $\omega$  is the angle of incidence and the angle of reflection.

## 7. THE OCCURRENCE PROBABILITY

Following Cox and Munk, let  $P$  stand for the probability

$$P \equiv p(\xi_x, \xi_y, W) d\xi_x d\xi_y\tag{6}$$

that a wave facet will occur with a slope within  $\pm d\xi_x/2$  of  $\xi_x$  and  $\pm d\xi_y/2$  of  $\xi_y$  when the wind speed is  $W$ . The wave slope occurrence probability density,  $p$ , is proportional to the horizontal projection of the facet. Cox and Munk obtained an expression for  $p$  whose lowest order term is

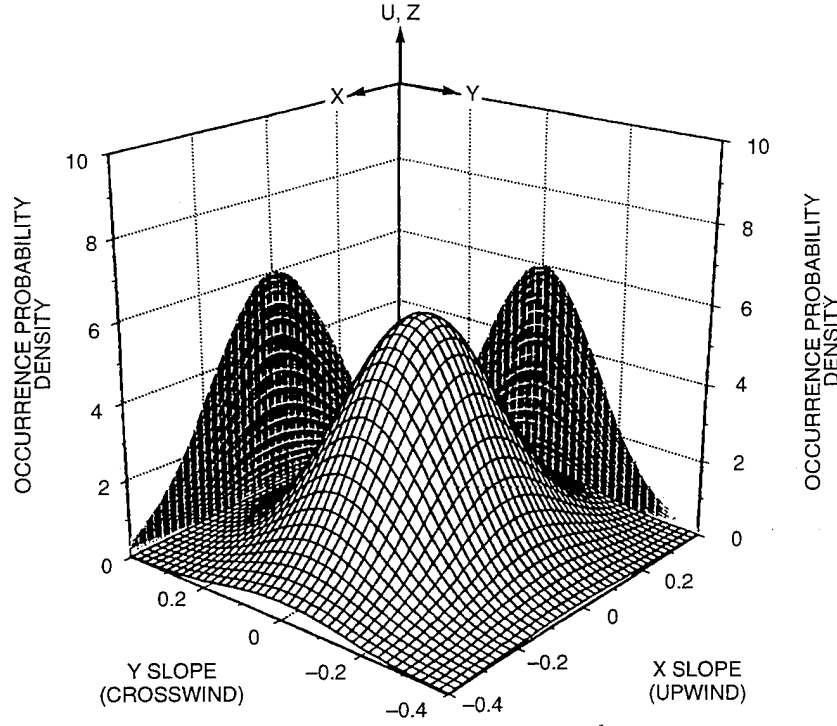
$$\begin{aligned}p(\xi_x, \xi_y, W) &\approx \frac{1}{2\pi\sigma_u\sigma_c} \exp \left\{ -\frac{1}{2} \left( \frac{\xi_x^2}{\sigma_u^2} + \frac{\xi_y^2}{\sigma_c^2} \right) \right\} \\ \sigma_u^2 &= 0.000 + 3.16 \cdot 10^{-3}W \\ \sigma_c^2 &= 0.003 + 1.92 \cdot 10^{-3}W\end{aligned}\tag{7}$$

Here  $\sigma_u^2$  and  $\sigma_c^2$  are the variances in  $\xi_x$  and  $\xi_y$  respectively and  $W$  is the wind speed in  $\text{m s}^{-1}$ . Figure 3 shows the dependence of  $p$  throughout slope space for a wind speed of  $10 \text{ m s}^{-1}$ . The coordinate system of figure 2 has been inserted at the top of the figure to illustrate the relation between coordinates and slopes. Note that the first  $X$ - $Y$  quadrant corresponds to negative slopes.

---

6. The zenith angle of  $\mathbf{U}_n$  is the same as the tilt of the facet. The tilt is the angle of the steepest ascent within the facet.





**Figure 3.** Cox-Munk occurrence probability density for a windspeed of  $10 \text{ m s}^{-1}$ .

## 8. THE INTERACTION PROBABILITY

Following a suggestion of Plass, et al. (1976), let  $Q$  stand for the (different) probability

$$Q \equiv q(\xi_x, \xi_y, \theta, \phi, W) \xi_x d\xi_y \quad (8)$$

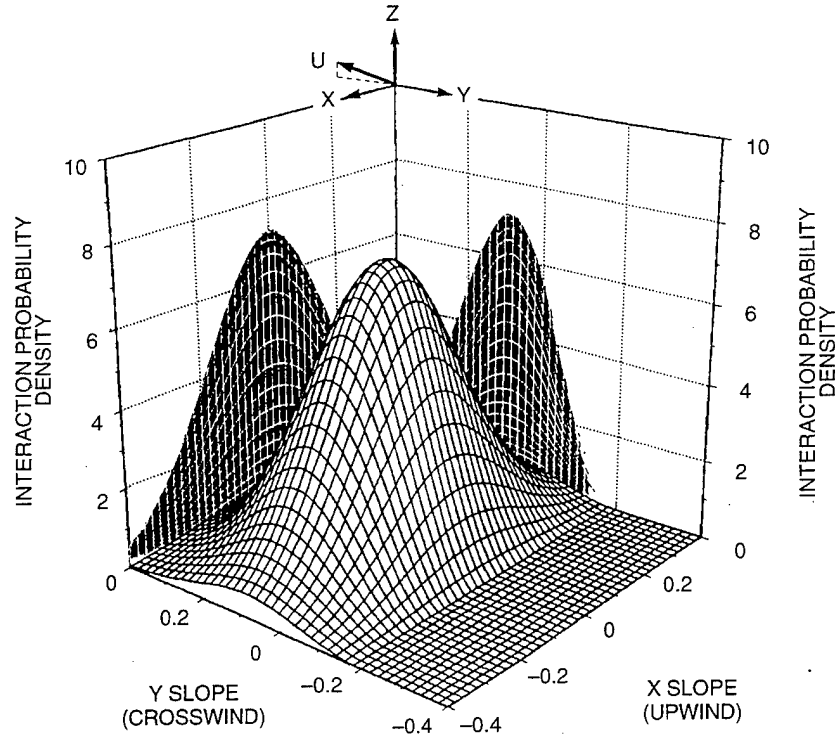
that a facet whose slope is within  $\pm d\xi_x/2$  of  $\xi_x$  and  $\pm d\xi_y/2$  of  $\xi_y$  will interact with a ray arriving from the arbitrary direction  $\mathbf{U} = (\theta, \phi)$  when the wind speed is  $W$ . The wave slope interaction probability density,  $q$ , is proportional to the facet area projected normal to the ray. It has previously been shown (Zeisse, 1994, 1995)<sup>7</sup> that

$$q(\xi_x, \xi_y, \theta, \phi, W) = \frac{\frac{\cos \omega}{\cos \theta_n} p}{\iint_{\substack{\omega \leq \pi/2 \\ U = \text{const.}}} \frac{\cos \omega}{\cos \theta_n} p d\xi_x d\xi_y} \quad (9)$$

Figure 4 is a graph of equation (9), also for a wind speed of  $10 \text{ m s}^{-1}$ , showing how facets with a specified slope interact with a ray pointing in the direction  $(80^\circ, 270^\circ)$ .

7. Equation (9) is only defined for  $\omega \leq \frac{\pi}{2}$ .





**Figure 4.** Cox-Munk-Plass interaction probability density for a windspeed of  $10 \text{ m s}^{-1}$ .

## 9. EQUATIONS FOR SEA RADIANCE

The capillary wave contributions to sea radiance are

$$N_{sky}(\theta_r, \phi_r, W, \nu) = \int \int_{\substack{\theta_s, \omega \leq \pi/2 \\ U_r = \text{const.}}} N_s(\theta_s, \phi_s, \nu) \rho(\omega, \nu) q(\xi_x, \xi_y, \theta_r, \phi_r, W) d\xi_x d\xi_y \quad (10)$$

$$N_{sun}(\theta_o, \phi_o, \theta_r, \phi_r, W, \nu) \approx \frac{N_o(\theta_o, \phi_o, \nu)}{4} \quad (11)$$

$$\int \int_{\substack{\text{disk} \\ U_r = \text{const.}}} \rho(\omega, \nu) \sec \omega \sec^3 \theta_n q(\xi_x, \xi_y, \theta_r, \phi_r, W) \sin \theta_s d\theta_s d\phi_s$$

$$N_{sea}(\theta_r, \phi_r, W, T_{sea}, \nu) = N_{bb}(T_{sea}, \nu) \int \int_{\substack{\omega \leq \pi/2 \\ U_r = \text{const.}}} [1 - \rho(\omega, \nu)] q(\xi_x, \xi_y, \theta_r, \phi_r, W) d\xi_x d\xi_y \quad (12)$$

In each of the integrals (10) through (12),  $q$  plays the role of a weighting function attached to the facet. The weight is applied to the ray leaving the footprint and propagating toward the receiver, and



that ray and those receiver coordinates are held constant in all of the integrals. A physical description and some mathematical details of each equation will now be presented.

In the integrand of (10), the product of  $N_s$  and  $\rho$  represents the radiance leaving a single facet when  $N_s(\theta_s, \phi_s, \nu)$  is the spectral sky radiance incident on that facet at zenith angle  $\theta_s$  and azimuth  $\phi_s$ . This product is weighted by  $q$  and integrated over all slopes in the ocean. During integration, a specular reflection occurs at one facet after another inside the footprint with the outgoing (receiver) ray held fixed. The source ray is swept across the sky and sun. Equation (10) will require explicit expressions for each of its variables in terms of slopes and receiver coordinates. From (2) and (4) it can be shown that the facet tilt is given in terms of the facet slopes by

$$\cos \theta_n = c_n = \frac{1}{\sqrt{1 + \xi_x^2 + \xi_y^2}} \quad (13)$$

while the fact that  $\omega$  is the angle between the facet normal and the receiver ray implies that

$$\begin{aligned} \cos \omega &= \mathbf{U}_n \cdot \mathbf{U}_r \\ &= a_n a_r + b_n b_r + c_n c_r \\ &= \{-\xi_x a_r - \xi_y b_r + c_r\} c_n \\ &= \frac{\{-\xi_x \sin \theta_r \cos \phi_r - \xi_y \sin \theta_r \sin \phi_r + \cos \theta_r\}}{\sqrt{1 + \xi_x^2 + \xi_y^2}} \end{aligned} \quad (14)$$

Equations (13) and (14) hold at all times, regardless of whether a specular reflection is taking place. When a specular reflection does occur, the Z component of the law of reflection

$$\mathbf{U}_s = 2 \cos \omega \mathbf{U}_n - \mathbf{U}_r \quad (15)$$

gives

$$\begin{aligned} \cos \theta_s &= 2 \cos \omega c_n - c_r \\ &= \frac{2\{-\xi_x a_r - \xi_y b_r + c_r\} - c_r/c_n^2}{1/c_n^2} \\ &= \frac{-2 \sin \theta_r (\xi_x \cos \phi_r + \xi_y \sin \phi_r) + \cos \theta_r (1 - \xi_x^2 - \xi_y^2)}{1 + \xi_x^2 + \xi_y^2} \end{aligned} \quad (16)$$

where  $\{ \}$  represents the expression within curly braces in (14). Finally, the X and Y components of (15) give

$$\begin{aligned} \tan \phi_s &= \frac{b_s}{a_s} \\ &= \frac{2 \cos \omega b_n - b_r}{2 \cos \omega a_n - a_r} \\ &= \frac{2\xi_y \{-\xi_x a_r - \xi_y b_r + c_r\} + b_r/c_n^2}{2\xi_y \{-\xi_x a_r - \xi_y b_r + c_r\} + a_r/c_n^2} \\ &= \frac{(1 + \xi_x^2 - \xi_y^2) \sin \phi_r - (2\xi_x \xi_y) \cos \phi_r + (2\xi_y) \cot \theta_r}{(1 - \xi_x^2 + \xi_y^2) \cos \phi_r - (2\xi_x \xi_y) \sin \phi_r + (2\xi_x) \cot \theta_r} \end{aligned} \quad (17)$$

for the source azimuth during specular reflection by a facet  $(\xi_x, \xi_y)$  into a receiver at  $(\theta_r, \phi_r)$ .



Equations (13), (14), (16), and (17) should be used in (10) [and in equation(9) when using (10)]. The Cartesian expressions are convenient for computer calculation while the spherical expressions are consistent with the form of equations (10) through (12).

In the integrand of (11), the product of  $N_o$  and  $\rho$  represents the spectral radiance leaving a single facet when  $N_o(\theta_o, \phi_o, \nu)$  is the spectral radiance arriving at that facet from the sun whose center is at  $(\theta_o, \phi_o)$ . The remaining factors in (11), apart from  $q$ , are the Jacobian of the transformation from ocean slopes to sky coordinates (Zeisse, 1994). As before, the integrand is weighted by  $q$  but now the integration is over the solar disk in the sky. (It is assumed in (11) that  $N_o(\theta_o, \phi_o, \nu)$  does not vary during integration because the sun is a Lambertian source and the solar disk is small.) During integration, a specular reflection from the sun to the receiver occurs at those facets within the footprint with the correct slope. Explicit expressions for each of the variables in terms of source and receiver coordinates will be required in (11). The law of reflection

$$2 \cos \omega \mathbf{U}_n = \mathbf{U}_s + \mathbf{U}_r \quad (18)$$

gives the facet position in terms of the source and receiver positions whenever a specular reflection occurs. The components of (18) give

$$\begin{aligned} \xi_x &= -\frac{a_n}{c_n} \\ &= -\frac{a_s + a_r}{c_s + c_r} \\ &= -\frac{\sin\theta_s \cos\phi_s + \sin\theta_r \cos\phi_r}{\cos\theta_s + \cos\theta_r} \end{aligned} \quad (19)$$

and

$$\begin{aligned} \xi_y &= -\frac{b_n}{c_n} \\ &= -\frac{b_s + b_r}{c_s + c_r} \\ &= -\frac{\sin\theta_s \sin\phi_s + \sin\theta_r \sin\phi_r}{\cos\theta_s + \cos\theta_r} \end{aligned} \quad (20)$$

while its square gives

$$\begin{aligned} 2 \cos^2 \omega &= 1 + \mathbf{U}_s \cdot \mathbf{U}_r \\ &= 1 + a_s a_r + b_s b_r + c_s c_r \\ &= 1 + \sin\theta_s \sin\theta_r \cos(\phi_s - \phi_r) + \cos\theta_s \cos\theta_r \end{aligned} \quad (21)$$



Finally, from (2), (4), and (18) we have

$$\begin{aligned}
 \tan^2 \theta_n &= \xi_x^2 + \xi_y^2 \\
 &= \frac{(a_s + a_r)^2 + (b_s + b_r)^2}{(c_s + c_r)^2} \\
 &= \frac{\sin^2 \theta_s + \sin^2 \theta_r + 2 \sin \theta_s \sin \theta_r \cos(\phi_s - \phi_r)}{(\cos \theta_s + \cos \theta_r)^2}
 \end{aligned} \tag{22}$$

Expressions (19) through (22) should be used in (11) [and in (9) when using (11)]. They apply only when there is a specular reflection.

In (12) there is no incident ray or specular reflection, and integration is over all slopes in the ocean. The integral in (12) is the effective spectral emissivity of the ocean. Explicit expressions in terms of slopes and receiver coordinates will also be required for each of the variables in (12) [and in (9) when using (12)]. Equation (13) is the expression for  $\theta_n$  and equation (14) is the expression for  $\omega$ .

## 10. SEARAD

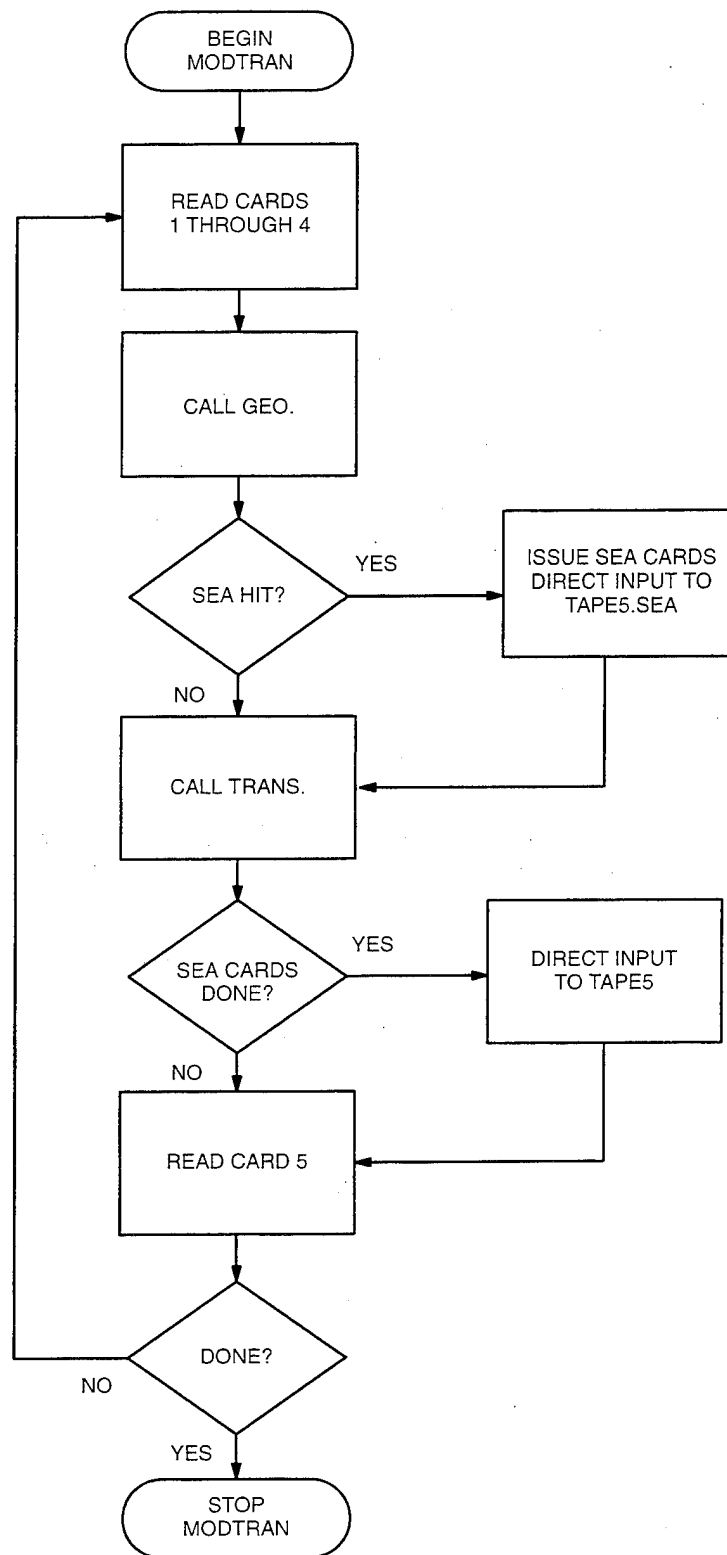
*SeaRad* consists of new routines added to MODTRAN2 to compute the spectral values of  $N_{sky}$ ,  $N_{sun}$ , and  $N_{sea}$  according to equations (10), (11), and (12), respectively. Through modifications to subroutine "TRANS", these values are assembled according to (1) and integrated over the wave number after obtaining proper path radiance and transmittance spectral values. *SeaRad* also introduces minor changes in subroutine "DPFNMN" and major changes in subroutine "DRIVER". These changes will now be considered in more detail.

The modifications to "DRIVER" are briefly shown in figure 5. After the normal call to "GEO", a test is conducted to see whether the ray chosen by the user has hit the surface of the sea. If so, geometry cards required for the sea calculation are issued by subroutine "Card" to file "Tape5.Sea", and input is temporarily redirected to "Tape5.Sea". An example of "Tape5.Sea" is given on page A-4, for the run initiated by the input file on page A-2. After the final card has been read from "Tape5.Sea", sea radiance is calculated in "TRANS". Then "TAPES" is restored as the active input file and normal program execution is resumed. Please see Appendix B for a detailed flowchart of the modifications to "DRIVER" as well as the complete source code for the modified version of "DRIVER".

Conditions in "DPFNMN" determine whether or not the sea has been hit. "DPFNMN" is a subroutine reached by a sequence of calls beginning in the driver with a call to subroutine "GEO". Modifications to "DPFNMN" are summarized in figure 6. A logical variable "Sea", initially set false, is set true in "DPFNMN" if the following four conditions are met:

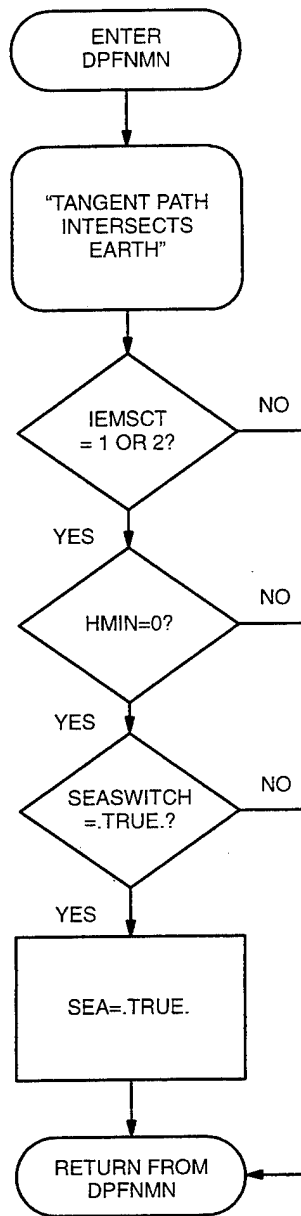
1. The program has reached the section of code following the comment line "Tangent path intersects earth."
2. The user has chosen a radiance mode.
3. The variable "HMIN" is equal to zero.
4. The variable "SeaSwitch" is true.





**Figure 5.** Flowchart for modifications to MODTRAN2 subroutine "DRIVER."



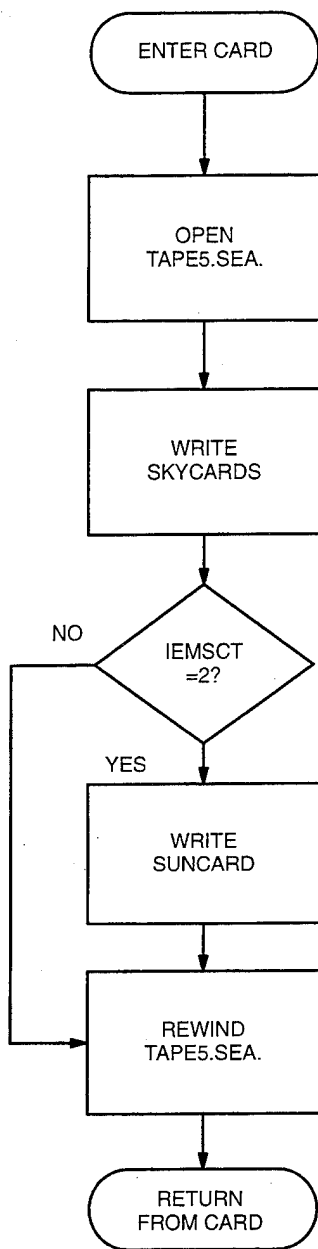


**Figure 6.** Flowchart for modifications to MODTRAN2 subroutine "DPFNMN."

The variable "Sea" is stored in a common block made available to the driver, which inspects "Sea" before and after each of its calls to "GEO". A change from false to true indicates that the ocean has been hit during that call. A hit induces a geometry calculation by a call to subroutine "Foot" (if IEMSCT = 1) or subroutine "SunFoot" (if IEMSCT = 2). This is followed in each case by a call to subroutine "Card".

The purpose of "Card" is to supply sources for the Cox-Munk routines "Sky" and "Sun." As shown in figure 7, geometry cards are issued here to file "Tape5.Sea" to obtain spectral radiance along paths to the sky at three separate zenith angles. These three cards, one for each zenith angle, are called "Sky Cards" in the flowchart. Later these data will be used by subroutine "Fit" to establish a two-parameter least squares fit at each wave number providing "Sky" with the sky dome radiance.





**Figure 7.** Flowchart for SeaRad subroutine "Card."



If the sun is involved, "Card" will issue a fourth and final Card 3, called a "Sun Card" in the flow-chart, which provides solar irradiance for later use as a source by subroutine "Sun".

The modifications described to this point have, in effect, inserted three sky cards (followed by a sun card if necessary) into the user's input file without the user's knowledge. The insertion is made only if the user has chosen a Card 3 whose path terminates on the surface of the earth. Such a Card 3 is called a "Path Card" in figure C-1. Within the wave number integration loop in "TRANS", spectral values of transmission, incident sky radiance, and incident solar irradiance are stored in arrays  $\text{Tau}(V)$ ,  $\text{Nsky}(V)$ , and  $\text{Ho}(V)$ , respectively. Outside the wave number integration loop these values are recalled for the sea radiance calculation by subroutine "Sky" (or subroutine "Sun" if  $\text{IEMSCT} = 2$ ).

The modified version of "DRIVER" is contained in Appendix B along with a detailed flowchart of its modifications. Appendix C contains the source code and a flowchart for the modified version of "TRANS", and Appendix D contains new code for the sea radiance calculation.

## 11. CONCLUSION

*SeaRad*, a modification of MODTRAN2, computes sea radiance between  $52.63 \text{ cm}^{-1}$  and  $25000 \text{ cm}^{-1}$ . Preliminary comparisons with data show that *SeaRad* has an approximate accuracy of several °C in the infrared.

*SeaRad* is currently designed for a single pixel and takes approximately 10 s to execute. Each time a new geometry is chosen by the user, *SeaRad* recalculates the source radiance and the path radiance and transmission. However, only the path properties change significantly from one pixel to the next in a typical ocean image. If *SeaRad* were redesigned to apply to sea images, the speed per pixel could be reduced, up to a factor of almost four, by calculating values of source radiance just once for the entire image.



## 12. REFERENCES

- Berk, A., L. S. Bernstein, and D. C. Robertson. 1989. "MODTRAN: A Moderate Resolution Model for LOWTRAN 7," Report GL-TR-89-0122. Air Force Geophysics Laboratory, Bedford, MA.
- Cox, C. and W. Munk. 1954. "Measurement of the Roughness of the Sea Surface from Photographs of the Sun's Glitter," *Journal of the Optical Society of America* 44: 838-850.
- Cox, C. and W. Munk. 1956. "Slopes of the Sea Surface Deduced from Photographs of Sun Glitter," *Scripps Institution of Oceanography Bulletin* 6: 401-487.
- Fenn, R. W., S. A. Clough, W. O. Gallery, R. E. Good, F. X. Kneizys, J. D. Mill, L. S. Rothman, E. P. Shettle, and F. E. Volz. 1985. "Optical and Infrared Properties of the Atmosphere," In *Handbook of Geophysics and the Space Environment*. A. S. Jursa, Ed. Air Force Geophysics Laboratory, Bedford, MA.
- Hale, G. M. and M. R. Querry. 1973. "Optical Constants of Water in the 200 nm to 200  $\mu$ m Wavelength Region," *Applied Optics* 3: 555-563.
- Isaacs, R. G., W. C. Wang, R. D. Worsham, and S. Goldenberg. 1986. "Multiple Scattering Treatment for Use in the LOWTRAN and FASCODE Models," Report AFGL-TR-86-0073: Air Force Geophysics Laboratory, Bedford, MA.
- Kneizys, F. X., E. P. Shettle, L. W. Abreu, J. H. Chetwynd, G. P. Anderson, W. O. Gallery, J. E. A. Selby, and S. A. Clough. 1988. "Users Guide to LOWTRAN 7," Report AFGL-TR-88-0177. Air Force Geophysics Laboratory, Bedford, MA.
- Kneizys, F. X., E. P. Shettle, W. O. Gallery, J. H. Chetwynd, L. W. Abreu, J. E. A. Selby, S. A. Clough, and R. W. Fenn. 1983. "Atmospheric Transmittance/Radiance: Computer Code LOWTRAN 6," Report AFGL-TR-83-0187. Air Force Geophysics Laboratory, Bedford, MA.
- Kneizys, F. X., E. P. Shettle, W. O. Gallery, J. H. Chetwynd, L. W. Abreu, J. E. A. Selby, R. W. Fenn, and R. A. McClatchey. 1980. "Atmospheric Transmittance/Radiance: Computer Code LOWTRAN 5," Report AFGL-TR-80-0067. Air Force Geophysics Laboratory, Bedford, MA.
- Kneizys, F. X., J. E. A. Selby, J. H. Chetwynd, and R. A. McClatchey. 1978. "Atmospheric Transmittance/Radiance: Computer Code LOWTRAN 4," Report AFGL-TR-78-0053. Air Force Geophysics Laboratory, Bedford, MA.
- Plass, G. N., G. W. Kattawar, and J. A. Guinn. 1975. "Radiative Transfer in the Earth's Atmosphere and Ocean: Influence of Ocean Waves," *Applied Optics* 14: 1924-1936.
- Querry, M. R., W. E. Holland, R. C. Waring, L. M. Earls, and M. D. Querry. 1977. "Relative Reflectance and Complex Refractive Index in the Infrared for Saline Environmental Waters," *Journal of Geophysical Research* 82: 1425-1433.
- Selby, J. E. A., E. P. Shettle, and R. A. McClatchey. 1976. "Atmospheric Transmittance from 0.25 to 28.5  $\mu$ m: Supplement LOWTRAN 3B," Report AFGL-TR-76-0258. Air Force Geophysics Laboratory, Bedford, MA.
- Selby, J. E. A. and R. A. McClatchey. 1975. "Atmospheric Transmittance from 0.25 to 28.5  $\mu$ m: Computer Code LOWTRAN3," Report AFGL-TR-72-0255. Air Force Geophysics Laboratory, Bedford, MA.



- Selby, J. E. A. and R. A. McClatchey. 1972. "Atmospheric Transmittance from 0.25 to 28.5  $\mu\text{m}$ : Computer Code LOWTRAN 2," Report AFGL-TR-72-0745. Air Force Geophysics Laboratory, Bedford MA.
- Stratton, J. A. 1941. *Electromagnetic Theory*, pp. 505 ff., McGraw-Hill, New York, NY.
- Zeisse, C. R. 1994. "Radiance of the Ocean Horizon," NRaD TR 1660 (April). Naval Command, Control and Ocean Surveillance Center, RDT&E Division, San Diego, CA.
- Zeisse, C. R. 1995. "Radiance of Ocean Horizon," *Journal of the Optical Society*, vol. A, pp. 2022-2030.



**APPENDIX A**  
***SeaRad* INPUT AND OUTPUT FILES**



# **"TAPE5RAD.STD" INPUT FILE**

C:\MOD2\TAPE5RAD.FIL 7/20/95

F	6	3	1	1	0	0	0	0	0	0	0	0	288.15	0.00
	3	0	0	5	0	0	10.000	10.000	10.000				.000	.000
	00.023			.000	100.000		.000	.000	0.00	0		90.000	T	
		940		950	10		5							
	0													



# "OUT" FILE

C:\MOD2\OUT.FIL 7/20/95

\*\*\*\*\* SEARAD, A MODIFICATION OF LOWTRAN7 \*\*\*\*\*

DATE: 07/20/95

TIME: 15:11:38

THERMAL RADIANCE MODE

MULTIPLE SCATTERING USED

MARINE AEROSOL MODEL USED

WIND SPEED = 10.00 M/SEC  
WIND SPEED = 10.00 M/SEC, 24 HR AVERAGE  
RELATIVE HUMIDITY = 50.00 PERCENT  
AIRMASS CHARACTER = 5  
VISIBILITY = 10.00 KM

SLANT PATH TO SPACE

H1 = 0.023 KM  
HMIN = 0.000 KM  
ANGLE = 100.000 DEG

FREQUENCY RANGE

IV1 = 940 CM-1 ( 10.64 MICROMETERS)  
IV2 = 950 CM-1 ( 10.53 MICROMETERS)  
IDV = 10 CM-1  
IFWHM = 5 CM-1  
IFILTER = 0

SUMMARY OF THE GEOMETRY CALCULATION

H1 = 0.023 KM  
H2 = 0.000 KM  
ANGLE = 100.000 DEG  
RANGE = 0.133 KM  
BETA = 0.001 DEG  
PHI = 80.001 DEG  
HMIN = 0.000 KM  
BENDING = 0.000 DEG  
LEN = 0

SEA AT 288.15 K REPLACES BLACK BODY BOUNDARY

UPWIND = 90.000 DEG EAST OF LINE OF SIGHT

RECEIVED RADIANCE VALUES

PATH TO FOOTPRINT = 0.01814 W M-2 SR-1 (AV. TRANS. 0.9776)  
SEA EMISSION = 0.70712 W M-2 SR-1  
SKY REFLECTION = 0.06125 W M-2 SR-1  
SUN GLINT = 0.00000 W M-2 SR-1  
  
TOTAL RADIANCE = 0.78652 W M-2 SR-1  
BLACK BODY TEMP. = 6.7 C



# "TAPE5.SEA" FILE

C:\MOD2\TAPE5SEA.FIL 7/20/95

3	0.000	0.000	57.296	0.000	0.000	0.000	0	90.000	T
3	0.000	0.000	73.148	0.000	0.000	0.000	0	90.000	T
3	0.000	0.000	89.000	0.000	0.000	0.000	0	90.000	T



**APPENDIX B**  
**MODIFIED SUBROUTINE "DRIVER"**



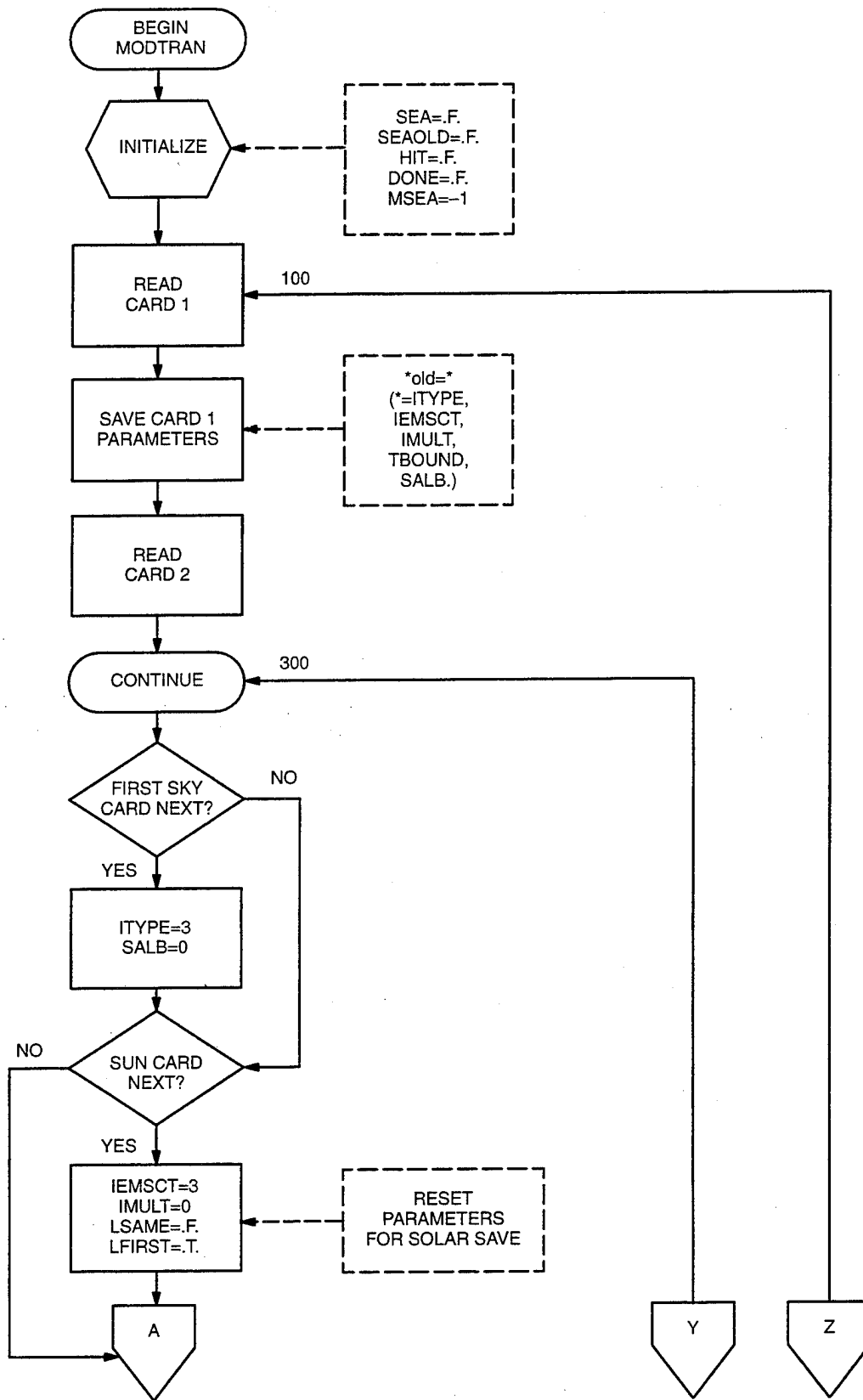


Figure B-1. Detailed flowchart for modified subroutine "DRIVER".



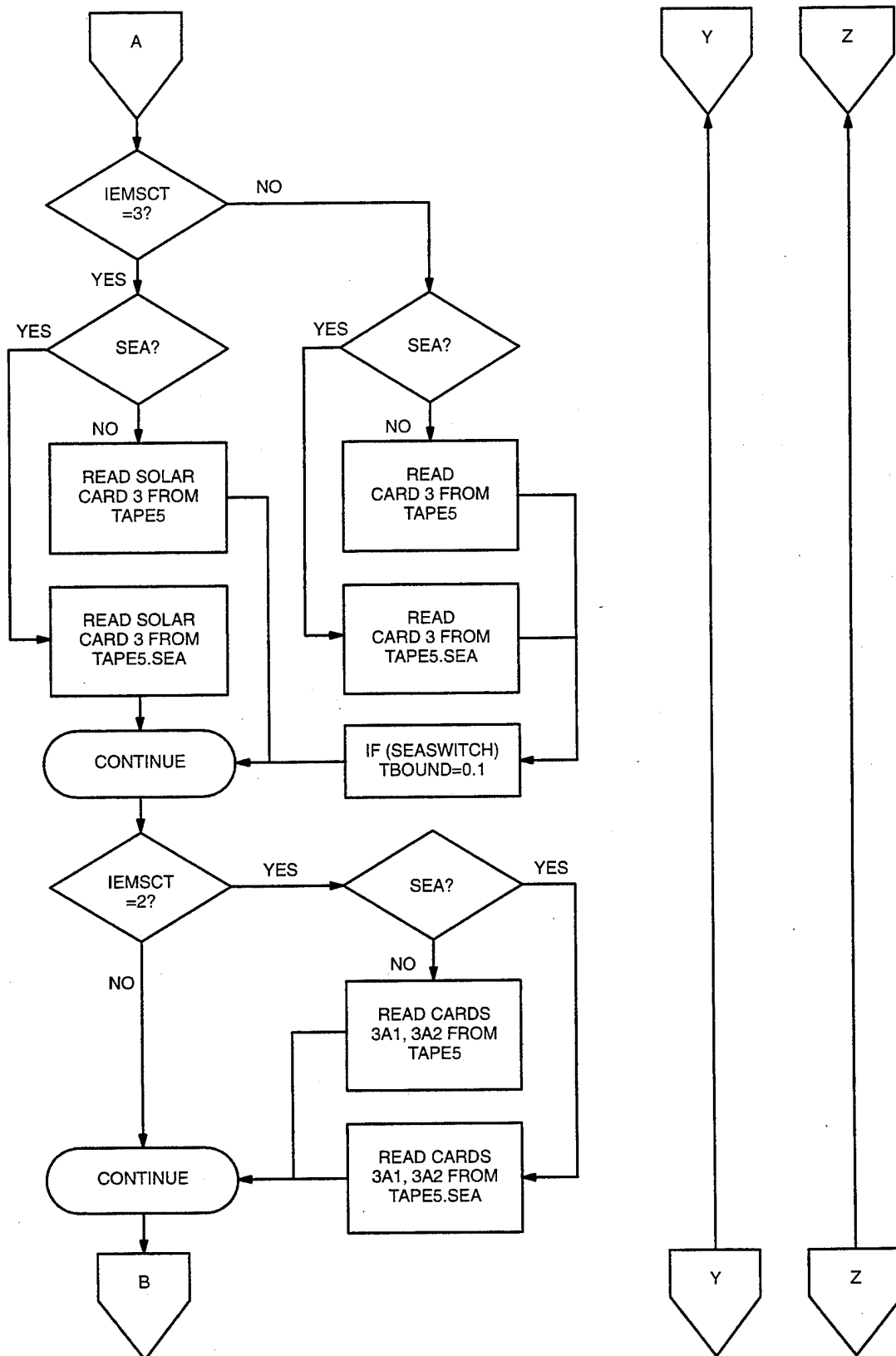


Figure B-1. Detailed flowchart for modified subroutine "DRIVER". (Continued)



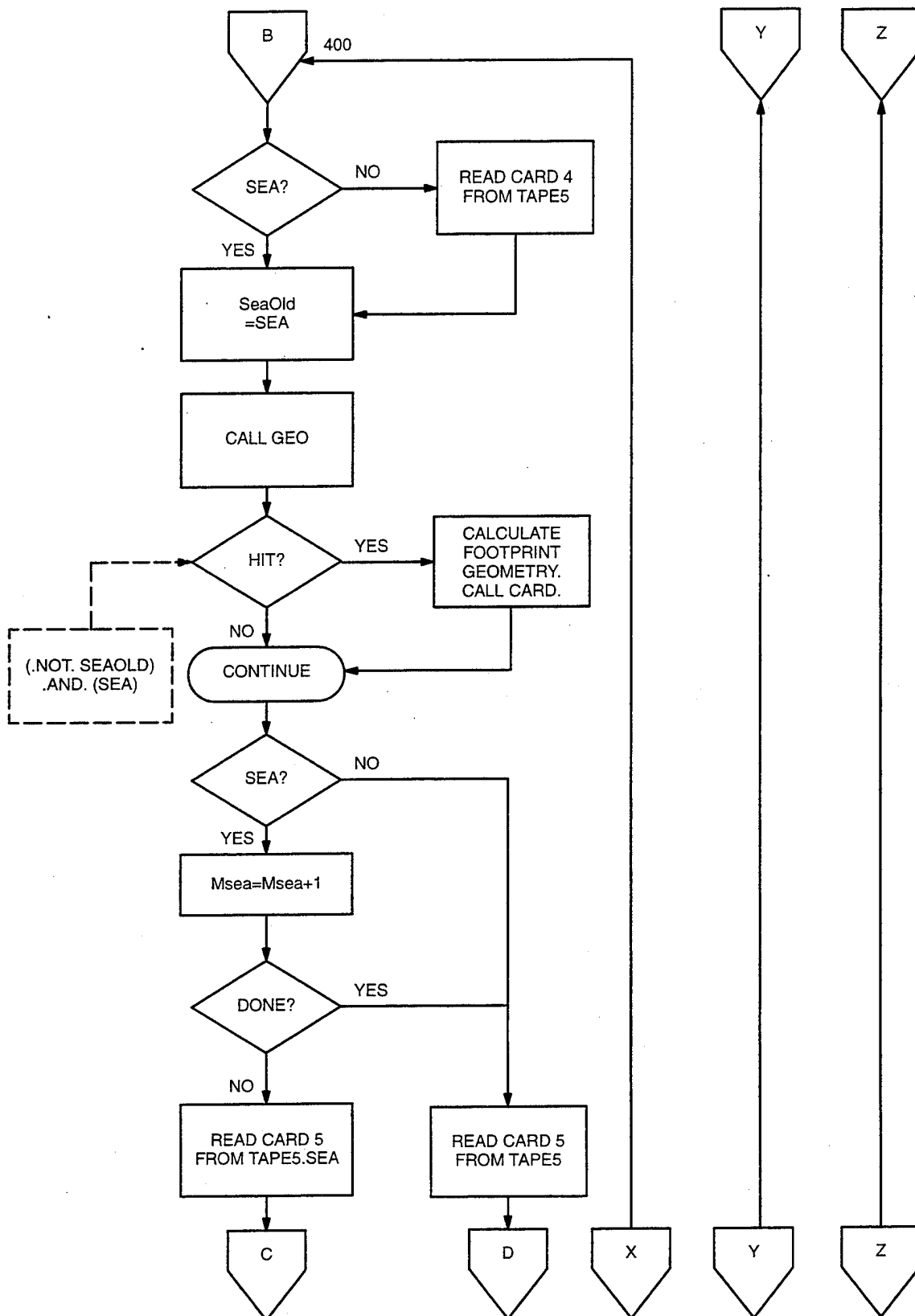
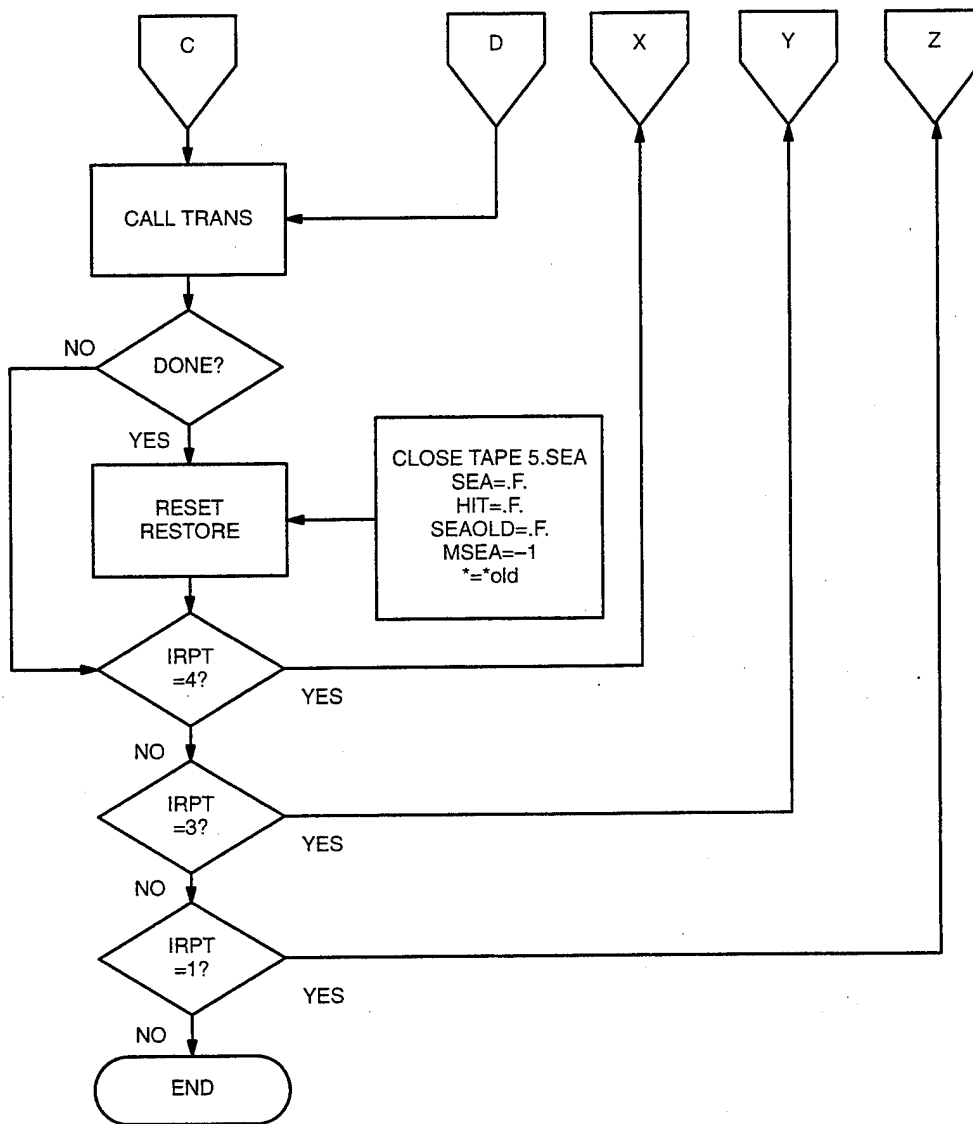


Figure B-1. Detailed flowchart for modified subroutine "DRIVER". (Continued)





**Figure B-1.** Detailed flowchart for modified subroutine "DRIVER". (Continued)



	SUBROUTINE DRIVER	driv 100
	COMMON RELHUM(34),HSTOR(34),ICH(4),VH(17),TX(63),W(63)	driv 110
C	COMMON WPATH(68, 63),TBBY(68)	driv 120
	COMMON IMSMX,WPATH(102,63),TBBY(102),PATM(102),NSPEC,KPOINT(12)	driv 130
	COMMON ABSC(5,47),EXTC(5,47),ASYM(5,47),VX2(47),AWCCON(5)	driv 140
	COMMON /IFIL/ IRD,IPR,IPU,NPR,IPR1,IP6,IP7,IP8,IP4,IRDS,IP6S,ITR,	
+	Isky,Isun,Ipath	
	COMMON /CARD1/ MODEL,ITYPE,IEMSCT,M1,M2,M3,IM,NOPRT,TBOUND,SALB	driv 160
1	,MODTRN	driv 170
	LOGICAL MODTRN	driv 180
	logical ground	driv 190
	logical lsame	driv 200
	LOGICAL SeaSwitch,Sea,SeaOld,Hit,Done	
	COMMON /CARD1A/ M4,M5,M6,MDEF,IRD1,IRD2	driv 210
	COMMON /CARD2/ IHAZE,ISEASN,IVULCN,ICSTL,ICLD,IVSA,VIS,WSS,WHH,	driv 220
1	RAINRT	driv 230
	COMMON /CARD2A/ CTHIK,CALT,CEXT	driv 240
	COMMON /CARD2D/ IREG(4),ALTB(4),IREGC(4)	driv 250
	COMMON /CARD3/ H1,H2,ANGLE,RANGE,BETA,RE,LEN,Psi,SeaSwitch	
	COMMON /Card3A1/ IPARM,IPH,IDAY,ISOURC	
	COMMON /Card3A2/ PARM1,PARM2,PARM3,PARM4,GMT,PSIPO,ANGLEM,G	
	COMMON /CARD4/ IV1,IV2,IDV,IFWHM,IFILTER	
	COMMON /CNSTNS/ PIX,CA,DEG,GCAIR,BIGNUM,BIGEXP	driv 290
	COMMON /CNTRL/ KMAX,M,IKMAX,NL,ML,IKLO,ISSGEO,IMULT	driv 300
	COMMON /MODEL/ ZM(34),PM(34),TM(34),RFNDX(34),DENSTY(63,34),	driv 310
1	CLDAMT(34),RRAMT(34),EQLWC(34),HAZEC(34)	driv 320
	COMMON /SOLS/ AH1(68),ARH(68),	driv 330
1	WPATHS(102,63),PA(68),PRX(68),ATHETA(35),ADBETA(35),LJ(69),	
2	JTURN,ANGSUN,CSZEN(68),TBBYS(102,12),PATMS(102,12)	driv 370
	COMMON /MART/ RHH	driv 380
	COMMON /USRDTA/ NANGLS,ANGF(50),F(4,50)	driv 390
	COMMON /MDLZ/ HMDLZ(8)	driv 400
	COMMON /ZVSALY/ ZVSA(10),RHVSA(10),AHVSA(10),IHVSA(10)	driv 410
	CHARACTER*4 HHAZE,HSEASN,HVULCN,BLANK,HMET,HMODEL,HTRRAD	driv 420
	COMMON /TITL/ HHAZE(5,16),HSEASN(5,2),HVULCN(5,8),BLANK,	driv 430
1	HMET(5,2),HMODEL(5,8),HTRRAD(6,4)	driv 440
	COMMON /VSB/ VSB(10)	driv 450
C	COMMON /MNL/ TBBSS(68),TBBMS(34),WPMS(34,63),IMSMX,WPMSS(34,63)	driv 460
	COMMON /PATH/PL(68),QTHETA(68),ITEST,HI,HF,AHT(68),tph(68)	driv 470
	COMMON /AERTM/TAE7,TAE12,TAE13,TAE14,TAE16	driv 480
	common /graund/gndalt	driv 490
	common /small3/small	driv 500
	common /solar/lsame	driv 510
	PARAMETER (Kr = 216, Kv = 400)	
	COMMON /Filters/ FLIST(5,6),	
+	FILTER1(45), BB1(Kr), FILTER2(54), BB2(Kr),	
+	FILTER3(39), BB3(Kr), FILTER4(47), BB4(Kr),	
+	FILTER5(101),BB5(Kr), FILTER6(75), BB6(Kr)	
	COMMON /Constants/ pi,r2d,d2r,epsilon,delta,onem,onem,infinit	
	COMMON /Geometry/ Tsun,Psun,Tr,Pr	
	COMMON /Sea/ Sea,Hit,Msea,TBOUNDold,IEMSCTold	
	COMMON /SeaIndex/ Alpha01(100), Alpha02(20),	
+	Beta01 (100), Beta02 (20)	
	logical lfirst	driv 530
	data lfirst/.true./	driv 540
	REAL infinity	
	CHARACTER*8 Date\$, Time\$	



```

CHARACTER*14 Prog$
INTEGER*4 Istart, Iend
C      First, get starting time to measure total execution time:
CALL TIMER(Istart)
C
C      lfirst is true when first solar parameters are read in a series
C      of runs involving solar parameters.
C
C*****HDATE AND HTIME CARRY THE DATA AND TIME AND MUST BE DOUBLE
C*****PRECISION ON A 32 BIT WORD COMPUTER
C@      DOUBLE PRECISION HDATE,HTIME
      DIMENSION PLST(68),CSSENSV(68),QTHETS(68)
      DATA IRPT / 0 /
C*****IRD, IPR, AND IPU ARE UNIT NUMBERS FOR INPUT, OUTPUT, AND
C*****IPR1 = OUTPUT OF MOLECULAR TRANSMITTANCE
      DATA          MAXGEO / 68/
      small = 2.0
      IP4 = 14
      IRD = 5
      IPR = 6
      IP6 = 16
      IP6S= 26
      IPU = 7
      IP7 = 17
      IPR1= 8
      IP8 = 18
      IRDS = 29
      ITR = 30
      ISCRCH = 10
      ITM = 31
      Isky = 32
      Isun = 33
      Ipath = 34
      OPEN (IRD, FILE='TAPE5', STATUS='OLD')
      OPEN (IPR, FILE='TAPE6', STATUS='UNKNOWN')
      OPEN (IP6, FILE='OUT', STATUS='UNKNOWN')
      OPEN (IPU, FILE='TAPE7', STATUS='UNKNOWN')
      OPEN (IP7, FILE='TAPE7.PLT',STATUS='UNKNOWN')
      OPEN (IPR1,FILE='TAPE8', STATUS='UNKNOWN')
      OPEN (IP8, FILE='TAPE8.PLT',STATUS='UNKNOWN')
      OPEN (IP4, FILE='OUT.PLT', STATUS='UNKNOWN')
      OPEN (ITR, FILE='TRANS.PLT',STATUS='UNKNOWN')
      OPEN (ISCRCH,STATUS='SCRATCH',FORM='UNFORMATTED')
      OPEN (Isky, FILE='Sky.plt', STATUS='UNKNOWN')
      OPEN (Isun, FILE='Sun.plt', STATUS='UNKNOWN')
      OPEN (Ipath,FILE='Path.plt',STATUS='UNKNOWN')
      OPEN (ITM, FILE='TIME', STATUS='UNKNOWN')
C
C      ALTITUDE PARAMETERS
C
C      ZMDL COMMON/MODEL/ THE ALTITUDES USED IN LOWTRAN
C      ZCVSA,ZTVSA,ZIVSA CARD 3.3 LOWTRAN FOR VSA INPUT
C      ZVSA NINE ALTITUDES GEN BY VSA ROUTINE
C
      Pix=2.0*ASIN(1.0)
      CA=Pix/180.

```

```

driv 550
driv 560
driv 570
driv 580
driv 590
driv 600
driv 610
driv 620
driv 630
driv 640
driv 650
driv 660
driv 520
driv 670
driv 680
driv 690
driv 700
driv 710
driv 720
driv 730
driv 740
driv 750
driv 760
driv 770
driv 780
driv 790
driv 800
driv 810
driv 820
driv 830
driv 840
driv 850

```



```

DEG= 1.0/CA
pi      = Pix
r2d     = 180./pi
d2r     = pi/180.
epsilon = d2r*0.2659
delta   = 1.4E-6
onem    = 1. - delta
onep    = 1. + delta
infinity = 999999
RANGE=0.0
*****GCAIR IS THE GAS CONSTANT FOR AIR IN UNITS OF MB/(GM CM-3 K)
GCAIR = 2.87053E+3
*****BIGNUM AND BIGEXP ARE THE LARGEST NUMBER AND THE LARGEST ARGUMENT
*****EXP ALLOWED AND ARE MACHINE DEPENDENT. THE NUMBERS USED HERE ARE
*****A TYPICAL 32 BIT-WORD COMPUTER.
BIGNUM = 1.0E35
BIGEXP = 87.0
C      THE VALUES FOR BIGNUM AND BIGEXP FOLLOW THE
C      DESCRIPTION UNDER EXP FUNCTION IN "IBM SYSTEM 360/
C      AND SYSTEM 370 FORTRAN IV LANGUAGE"
C      BIGNUM = 4.3E68
C      BIGEXP = 174.6
KMAX=63
*****NL IS THE NUMBER OF BOUNDARIES IN THE STANDARD MODELS 1 TO 6
*****BOUNDARY 34 (AT 99999 KM) IS NO LONGER USED
NL = 33
*****
Sea     = .FALSE.
SeaOld  = .FALSE.
Hit     = .FALSE.
Msea    = -1
Done    = .FALSE.
*****
*****CALL TIME AND DATE:
*****THE USER MAY WISH TO INCLUDE SUBROUTINES FDATE AND FCLOCK WHICH
*****RETURN THE DATE AND TIME IN MM/DD/YY AND HH.MM.SS FORMATS
*****RESPECTIVELY. THE REQUIRED ROUTINES FOR A CDC 6600 ARE INCLUDED AT
*****THE MAIN PROGRAM IN COMMENT CARDS.
C@      CALL FDATE(HDATE)
C@      CALL FCLOCK(HTIME)
C@      CALL DATE (Date$)
C@      CALL TIME (Time$)
C
C*****START CALCULATION
C
C
100     DO 10 II = 1,4
10      IREG(II) = 0
        WRITE(IPR,1000)
1000    FORMAT('1',20X,'***** MODTRAN *****')
C@      WRITE(IPR,1010) HDATE,HTIME
1010    FORMAT('1',20X,'***** MODTRAN *****',10X,2(1X,A8,1X))
        DO 80 I=1,4
          DO 80 J=1,40
            ABSC(I,J)=0.
            EXTC(I,J)=0.
80      ASYM(I,J)=0.

```

driv 860

driv 870

driv 880

driv 890

driv 900

driv 910

driv 920

driv 930

driv 940

driv 950

driv 960

driv 970

driv 980

driv 990

driv1000

driv1010

driv1020

driv1030

driv1040

driv1050

driv1060

driv1070

driv1080

driv1090

driv1100

driv1110

driv1120

driv1130

driv1140

driv1150

driv1160

driv1170

driv1180

driv1190

driv1200

driv1210

driv1220

driv1230

driv1240

driv1250



```

JPRT = 0
IKLO=1
C
C*****CARD 1
C
      READ(IRD, '(L1,I4,12I5,F8.3,F7.2)')MODTRN,MODEL,ITYPE,
      + IEMSCT,IMULT,M1,M2,M3,M4,M5,M6,MDEF,IM,NOPRT,TBOUND,SALB
1110 FORMAT(13I5,F8.3,F7.2)
***** Save parameters to restore them
      ITYPEold = ITYPE
      IEMSCTold = IEMSCT
      IMULTold = IMULT
      If (TBOUND .EQ. 0.) TBOUND = 0.1
      TBOUNDold = TBOUND
      SALBold = SALB
***** in case new geometry cards are
      later introduced via file TAPE5.SEA
C
      IF (MODTRN) THEN
        Prog$ = 'MODTRAN2 *****'
      ELSE
        Prog$ = 'LOWTRAN7 *****'
      END IF
      WRITE (IP6, 1018) Prog$
1018  FORMAT(15X, '***** SEARAD, A MODIFICATION OF ', A14)
      WRITE (IP6, 1020) Date$, Time$
1020  FORMAT (/, 'DATE:', 1X, A8, T60, 'TIME:', 1X, A8)
      SELECT CASE (IEMSCT)
        CASE (0)
          WRITE (IP6, '(/, 18HTRANSMITTANCE MODE)')
        CASE (1)
          WRITE (IP6, '(/, 21HTHERMAL RADIANCE MODE)')
        CASE (2)
          WRITE (IP6, '(/, 32HTHERMAL PLUS SOLAR RADIANCE MODE)')
        CASE (3)
          WRITE (IP6, '(/, 21HSOLAR IRRADIANCE MODE)')
      END SELECT
C
      SELECT CASE (IMULT)
        CASE (0)
          PRINT *, "IMULT = ", IMULT, ": BEWARE OF BEN-SHALOM"
          WRITE (IP6, '(/, 22HSINGLE SCATTERING USED)')
        CASE (1)
          WRITE (IP6, '(/, 24HMULTIPLE SCATTERING USED)')
      END SELECT
C
      WRITE(IPR, '(15H0 CARD 1 *****L1,I4,12I5,F8.3,F7.2)')MODTRN,MODEL
1  , ITYPE,IEMSCT,IMULT,M1,M2,M3,M4,M5,M6,MDEF,IM,NOPRT,TBOUND,SALB
1111 FORMAT('0 CARD 1 *****',13I5,F8.3,F7.2)
C      IF(IMULT .EQ. 1 .AND. NOPRT.EQ. 1) NOPRT = 0
C
C      SET THE NUMBER OF SPECIES TREATED WITH THE 1 CM-1 BAND MODEL.
C      ALSO, FOR EACH SPECIES, SET THE POINTER WHICH MAPS THE HITRAN
C      NUMERICAL LABEL TO THE LOWTRAN NUMERICAL LABEL.
C
C      NSPEC=12
      KPOINT( 1)=17
      KPOINT( 2)=36

```

```

driv1260
driv1270
driv1280
driv1290
driv1300
driv1340
driv1350
driv1380
driv1390
driv1400
driv1410
driv1420
driv1430
driv1440
driv1450
driv1460
driv1470
driv1480
driv1490

```



KPOINT( 3)=31	driv1500
KPOINT( 4)=47	driv1510
KPOINT( 5)=44	driv1520
KPOINT( 6)=46	driv1530
KPOINT( 7)=50	driv1540
KPOINT( 8)=54	driv1550
KPOINT( 9)=56	driv1560
KPOINT(10)=55	driv1570
KPOINT(11)=52	driv1580
KPOINT(12)=11	driv1590
C	driv1600
IRD1 = 0	driv1610
IRD2 = 0	driv1620
IF (MODEL.EQ.0) LEN = 0	driv1630
IF((MODEL.EQ.0) .OR. (MODEL.EQ.7)) GO TO 110	driv1640
IF(M1.EQ.0) M1=MODEL	driv1650
IF(M2.EQ.0) M2=MODEL	driv1660
IF(M3.EQ.0) M3=MODEL	driv1670
IF(M4.EQ.0) M4=MODEL	driv1680
IF(M5.EQ.0) M5=MODEL	driv1690
IF(M6.EQ.0) M6=MODEL	driv1700
IF(MDEF.EQ.0) MDEF=1	driv1710
110 CONTINUE	driv1720
M=MODEL	driv1730
NPR = NOPRT	driv1740
C*****CARD 2 AEROSOL MODEL	driv1750
READ(IRD,1200) IHAZE, ISEASN, IVULCN, ICSTL, ICLD, IVSA, VIS, WSS, WHH,	driv1760
1 RAINRT, GNDALT	driv1770
1200 FORMAT(6I5, 5F10.3)	driv1780
WRITE(IPR,1201) IHAZE, ISEASN, IVULCN, ICSTL, ICLD, IVSA, VIS, WSS, WHH,	driv1790
1 RAINRT, GNDALT	driv1800
IF(GNDALT.GT.0.) WRITE(IPR,1199) GNDALT	driv1810
1199 FORMAT(1H0, ' GNDALT =', F10.2)	driv1820
IF(GNDALT.GE.6.0) THEN	driv1830
WRITE(IPR,1202) GNDALT	driv1840
GNDALT=0.	driv1850
ENDIF	driv1860
1201 FORMAT('0 CARD 2 *****', 6I5, 5F10.3)	driv1870
1202 FORMAT('0 GNDALT GT 6.0 RESET TO ZERO, GNDALT WAS', F10.3)	driv1880
C	driv1890
IF(VIS.LE.0.0.AND.IHAZE.GT.0) VIS=VSB(IHAZE)	driv1900
RHH= 0.	driv1910
IF(MODEL.EQ.0.OR.MODEL.EQ.7) GO TO 205	driv1920
IF((MODEL.EQ.3.OR.MODEL.EQ.5).AND.ISEASN.EQ.0) ISEASN=2	driv1930
C	driv1940
IF(IVSA.EQ.1 .AND. IHAZE.EQ.3)	driv1950
1 CALL MARINE(VIS, MODEL, WSS, WHH, ICSTL, EXTC, ABSC, 1)	driv1960
ICH(1)=IHAZE	driv1970
ICH(2)=6	driv1980
ICH(3)=9+IVULCN	driv1990
205 IF(RAINRT.EQ.0) GO TO 210	driv2000
WRITE(IPR,1205) RAINRT	driv2010
1205 FORMAT('0 RAIN MODEL CALLED, RAIN RATE = ', F9.2, ' MM/HR')	driv2020
210 ICH(4)=18	driv2030
IF(ICH(1).LE.0) ICH(1)=1	driv2040
IF(ICH(3).LE.9) ICH(3)=10	driv2050
IF(ICLD.GE.1 .AND. ICLD.LE.11) THEN	driv2060



ICH(4)=ICH(3)	driv2070
ICH(3)=ICH(2)	driv2080
ICH(2)=ICLD	driv2090
END IF	driv2100
IFLGA=0	driv2110
IFLGT=0	driv2120
CTHIK=-99.	driv2130
CALT=-99.	driv2140
CEXT=-99.	driv2150
ISEED=-99	driv2160
IF(ICLD .LT. 18) GO TO 230	driv2170
C*****CARD 2A CIRRUS CLOUDS	driv2180
READ(IRD,1210)CTHIK,CALT,CEXT,ISEED	driv2190
1210 FORMAT(3F10.3,I10)	driv2200
WRITE(IPR,1211)CTHIK,CALT,CEXT,ISEED	driv2210
1211 FORMAT('O CARD 2A *****',3F10.3,I10)	driv2220
230 CONTINUE	driv2230
C*****CARD 2B VERTICAL STRUCTURE ALGORITHM	driv2240
ZCVSA=-99.	driv2250
ZTVSA=-99.	driv2260
ZINVSA=-99.	driv2270
C	driv2280
IF( IVSA. EQ. 0 ) GO TO 240	driv2290
READ (IRD,1230) ZCVSA,ZTVSA,ZINVSA	driv2300
1230 FORMAT(3F10.3)	driv2310
WRITE(IPR,1231)ZCVSA,ZTVSA,ZINVSA	driv2320
1231 FORMAT('O CARD 2B *****',3F10.3)	driv2330
C	driv2340
CALL VSA(IHAZE,VIS,ZCVSA,ZTVSA,ZINVSA,ZVSA,RHVSA,AHVSA,IHVSA)	driv2350
C	driv2360
C	driv2370
C	driv2380
240 IF (MODEL.NE.0 .AND. MODEL.NE.7 ) ML=NL	driv2390
MDELS=MODEL	driv2400
DO 250 I=1,5	driv2410
IF(MDELS.NE.0)HMODEL(I,7)=HMODEL(I,MDELS)	driv2420
250 IF(MDELS.EQ.0)HMODEL(I,7)=HMODEL(I,8)	driv2430
C	driv2440
IF(IM .EQ. 1) THEN	driv2450
IF((MODEL.EQ.7.AND.IM.EQ.1) .OR.(MODEL.EQ.0)) THEN	driv2460
C	driv2470
C*****CARD 2C USER SUPPLIED ATMOSPHERIC PROFILE	driv2480
C	driv2490
READ (IRD,1250) ML,IRD1,IRD2,(HMODEL(I,7),I=1,5)	driv2500
1250 FORMAT(3I5,18A4)	driv2510
WRITE(IPR,1251)ML,IRD1,IRD2,(HMODEL(I,7),I=1,5)	driv2520
IF(IVSA.EQ.1)CALL RDNSM	driv2530
1251 FORMAT('O CARD 2C *****',3I5,18A4)	driv2540
ENDIF	driv2550
ENDIF	driv2560
M=7	driv2570
CALL AERNM(JPRT, GNDALT)	driv2580
IF(ICLD .LT. 20) GO TO 260	driv2590
C	driv2600
C	driv2610
C	driv2620
IF(CTHIK.NE.0) IFLGT=1	driv2630



IF(CALT.NE.0) IFLGA=1	driv2640
IF(ISEED.EQ.0) IFLGT=2	driv2650
IF(ISEED.EQ.0) IFLGA=2	driv2660
CALL CIRRUS(CTHIK,CALT,ISEED,CPROB,CEXT)	driv2670
WRITE(IPR,1220)	driv2680
1220 FORMAT(15X,'CIRRUS ATTENUATION INCLUDED (N O A A CIRRUS) ')	driv2690
IF(IFLGT.EQ.0) WRITE(IPR,1221) CTHIK	driv2700
1221 FORMAT(15X,'CIRRUS ATTENUATION STATISTICALLY DETERMINED TO BE',	driv2710
1 F10.3,'KM')	driv2720
IF(IFLGT.EQ.1) WRITE(IPR,1222) CTHIK	driv2730
1222 FORMAT(15X,'CIRRUS THICKNESS USER DETERMINED TO BE',F10.3,'KM')	driv2740
IF(IFLGT.EQ.2) WRITE(IPR,1223) CTHIK	driv2750
1223 FORMAT(15X,'CIRRUS THICKNESS DEFAULTED TO MEAN VALUE OF ',	driv2760
1 F10.3,'KM')	driv2770
IF(IFLGA.EQ.0) WRITE(IPR,1224)CALT	driv2780
1224 FORMAT(15X,'CIRRUS BASE ALTITUDE STATISTICALLY DETERMINED TO BE',	driv2790
1 F10.3,' KM')	driv2800
IF(IFLGA.EQ.1) WRITE(IPR,1225) CALT	driv2810
1225 FORMAT(15X,'CIRRUS BASE ALTITUDE USER DETERMINED TO BE',	driv2820
1 F10.3,' KM')	driv2830
IF(IFLGA.EQ.2) WRITE(IPR,1226) CALT	driv2840
1226 FORMAT(15X,'CIRRUS BASE ALTITUDE DEFAULTED TO MEAN VALUE OF',	driv2850
1 F10.3,'KM')	driv2860
WRITE(IPR,1227)CPROB	driv2870
1227 FORMAT(15X,'PROBABILITY OF CLOUD OCCURRING IS',F7.1,' PERCENT')	driv2880
C	driv2890
C       END OF CIRRUS MODEL SET UP	driv2900
C	driv2910
260   CONTINUE	driv2920
C	driv2930
C	driv2940
C*****CARD 2E	driv2950
C	driv2960
IF((IHAZE.EQ.7).OR.(ICLD.EQ.11)) THEN	driv2970
C	driv2980
C*****   CARD 2E USER SUPPLIED AEROSOL EXTINCTION, ABSORPTION, AND	driv2990
C       ASYMMETRY	driv3000
C       CALL RDEXA	driv3010
C	driv3020
ENDIF	driv3030
300   CONTINUE	driv3040
	driv3050
IPARM =-99	driv3060
IPH    =-99	driv3070
IDAY   =-99	driv3080
ISOURC=-99	driv3090
C	driv3100
PARM1 =-99.	driv3110
PARM2 =-99.	driv3120
PARM3 =-99.	driv3130
PARM4 =-99.	driv3140
GMT    =-99.	
PSIPO = 0.	
ANGLEM=-99.	driv3170
G       =-99.	driv3180
C	driv3190
C*****CARD 3 GEOMETRY PARAMETERS	driv3200



```

C                                                                    driv3210
      IF ((SEA) .AND. (Msea .EQ. 0)) THEN
C          the first sky card is next.
C          Set emissivity to zero (TBOUND is already zero) and ITYPE to 3
C          for calculations coming up with cards in TAPE5.SEA:
          ITYPE = 3
          SALB = 0.0
      END IF
*** Set mode to sun irradiance (if a sun card will be next) *****
      IF ((Sea) .AND. (IEMSCTold .EQ. 2) .AND. (Msea .EQ. 3)) THEN
          IEMSCT = 3
          IMULT = 0
          LFIRST = .TRUE.
          LSAME = .FALSE.
      END IF
      IF (IEMSCT .EQ. 3) GO TO 315                                                                    driv3220
**** Read introduced geometry cards from file TAPE5.SEA *****
      IF (SEA) THEN
          READ(IRDS,1312)H1,H2,ANGLE,RANGE,BETA,RO,LEN,Psi,SeaSwitch
      ELSE
          READ (IRD,1312)H1,H2,ANGLE,RANGE,BETA,RO,LEN,Psi,SeaSwitch
      END IF
**** and remove the boundary (in sea AND sky) for a sea calculation ***
      IF (SeaSwitch) TBOUND = 0.1
1312  FORMAT(6F10.3,I5,F10.3,L5)                                                                    driv3240
      WRITE(IPR,1313)H1,H2,ANGLE,RANGE,BETA,RO,LEN,Psi,SeaSwitch
1313  FORMAT('0 CARD 3 *****',6F10.3,I5,F10.3,L5)                                                driv3260
      GO TO 320                                                                                          driv3270
C
C*****CARD 3 FOR DIRECTLY TRANSMITTED SOLAR RADIANCE (IEMSCT = 3)                                driv3280
C
      315 CONTINUE                                                                                      driv3290
**** Read introduced sun card from file TAPE5.SEA *****
      IF (Sea) THEN
          READ(IRDS,1316) H1,H2,ANGLE,IDAY,RO,ISOURC,ANGLEM
      ELSE
          READ(IRD, 1316) H1,H2,ANGLE,IDAY,RO,ISOURC,ANGLEM    driv3300
      END IF
*****
1316  FORMAT(3F10.3,I5,5X,F10.3,I5,F10.3)                                                            driv3310
      WRITE(IPR,1317) H1,H2,ANGLE,IDAY,RO,ISOURC,ANGLEM      driv3320
1317  FORMAT('0 CARD 3 *****',3F10.3,I5,5X,F10.3,I5,F10.3)  driv3330
      ITYPE = 3                                                driv3340
      RANGE = 0.0                                              driv3350
      BETA = 0.0                                               driv3360
      LEN = 0                                                  driv3370
C*****RO IS THE RADIUS OF THE EARTH                                                                    driv3380
320   RE=6371.23                                              driv3390
C          ***** ERRATA JULY 25                               driv3400
          IF(H1. LT. ZM(1) ) THEN                               driv3410
              WRITE(IPR,905) H1,ZM(1)                           driv3420
905   FORMAT(' H1 LESS THAN FIRST ALT RESET ',/              driv3430
X      ' H1 WAS ',F10.2,' 1ST ALT = ',F10.2)                  driv3440
              H1 = ZM(1)                                         driv3450
          ENDIF                                                  driv3460
C          ***** END ERRATA                                   driv3470
H1S = H1                                                        driv3480

```



H2S	= H2	driv3490
ANGLES	= ANGLE	driv3500
RANGS	= RANGE	driv3510
BETAS	= BETA	driv3520
ITYPES	=ITYPE	driv3530
LENS	= LEN	driv3540
IF (MODEL.EQ.0)	RO = RE	driv3560
IF (MODEL.EQ.1)	RE=6378.39	driv3570
IF (MODEL.EQ.4)	RE=6356.91	driv3580
IF (MODEL.EQ.5)	RE=6356.91	driv3590
IF (RO.GT.0.0)	RE=RO	driv3600
C		driv3610
	IF (IEMSCT.NE.2) GO TO 330	driv3620
C		driv3630
C*****CARD 3A1		driv3640
C		driv3650
	IF (SEA) THEN	
	READ(IRDS,1320) IPARM,IPH,IDAY,ISOURC	
	ELSE	
	READ(IRD,1320) IPARM,IPH,IDAY,ISOURC	
	END IF	
1320	FORMAT(4I5)	
	WRITE(IPR,1321) IPARM,IPH,IDAY,ISOURC	
1321	FORMAT('O CARD 3A1*****',4I5)	
C		driv3700
C*****CARD 3A2		driv3710
C		driv3720
	IF (SEA) THEN	
	READ(IRDS,1322) PARM1, PARM2, PARM3, PARM4,	
+	GMT,PSIPO,ANGLEM,G	
	ELSE	
	READ(IRD,1322) PARM1, PARM2, PARM3, PARM4,	
+	GMT,PSIPO,ANGLEM,G	
	END IF	
1322	FORMAT(8F10.3)	driv3740
	WRITE(IPR,1323) PARM1,PARM2,PARM3,PARM4,GMT,PSIPO,ANGLEM,G	
1323	FORMAT('O CARD 3A2*****',8F10.3)	driv3760
C		driv3770
CSSSISSSISSSISSSI	CHANGES BEGIN.	driv3780
C		driv3790
	REWIND(ISCRCH)	driv3800
C		driv3810
	IF (LFIRST .AND. IMULT .EQ. 1) THEN	driv3820
C		driv3830
C	SAVE SOLAR PARAMETERS FOR COMPARING LATER.	driv3840
C	NOTE THAT LFIRST IS TRUE AND IMULT (MULTIPLE SOLAR SCATTERING)	driv3850
	LFIRST = .FALSE.	driv3860
	CALL SVSOLA(IPARM,IPH,IDAY,ISOURC,PARM1,PARM2,PARM3,PARM4,	driv3870
\$	GMT,PSIPO,ANGLEM,	
\$	ISAVE1,ISAVE2,ISAVE3,ISAVE4,SAVE1,SAVE2,SAVE3,SAVE4,	driv3890
\$	SAVE5,SAVE6,SAVE7)	driv3900
	LSAME = .FALSE.	driv3910
C		driv3920
	ELSEIF (IMULT .EQ. 1 .AND. IRPT .EQ. 3) THEN	driv3930
C		driv3940
C	NOW COMPARE SOLAR PARAMETERS; LSAME IS TRUE IF THEY MATCH.	driv3950
	CALL COMPAR(IPARM,IPH,IDAY,ISOURC,PARM1,PARM2,PARM3,PARM4,	driv3960



\$	GMT,PSIPO,ANGLEM,	
\$	ISAVE1,ISAVE2,ISAVE3,ISAVE4,SAVE1,SAVE2,SAVE3,SAVE4,	driv3980
\$	SAVE5,SAVE6,SAVE7,LSAME)	driv3990
	CALL SVSOLA(IPARM,IPH,IDAY,ISOURC,PARM1,PARM2,PARM3,PARM4,	driv4000
\$	GMT,PSIPO,ANGLEM,	
\$	ISAVE1,ISAVE2,ISAVE3,ISAVE4,SAVE1,SAVE2,SAVE3,SAVE4,	driv4020
\$	SAVE5,SAVE6,SAVE7)	driv4030
	ELSE	driv4040
C		driv4050
C	GET READY FOR ANOTHER POSSIBLE FORTHCOMING SERIES OF MULTIPLE	driv4060
C	SOLAR SCATTERING RUNS.	driv4070
	LFIRST = .TRUE.	driv4080
	LSAME = .FALSE.	driv4090
	ENDIF	driv4100
C		driv4110
C	CSSISSISSISSISSI CHANGES END	driv4120
C		driv4130
	IF(IPH. EQ. 0) THEN	driv4140
	IF(G. GE. 1.0) G = .9999	driv4150
	IF(G. LE. -1.0) G = -.9999	driv4160
	ENDIF	driv4170
	IF (IPH.NE.1) GO TO 330	driv4180
C		driv4190
C	C*****CARD 3B1 USER DEFINED PHASE FUNCTION	driv4200
C		driv4210
C	C*****READ USER DEFINED PHASE FUNCTION	driv4220
C		driv4230
	READ(IRD,1326)NANGLS	driv4240
1326	FORMAT(I5)	driv4250
	WRITE(IPR,1327)NANGLS	driv4260
1327	FORMAT(' CARD 3B1*****',I5)	driv4270
C		driv4280
C	C*****CARD 3B2	driv4290
C		driv4300
	READ(IRD,1328)(ANGF(I),F(1,I),F(2,I),F(3,I),F(4,I),I=1,NANGLS)	driv4310
1328	FORMAT(5E10.3)	driv4320
	WRITE(IPR,1329)(ANGF(I),F(1,I),F(2,I),F(3,I),F(4,I),I=1,NANGLS)	driv4330
1329	FORMAT('O CARD 3B2*****',5E10.3)	driv4340
C		driv4350
	330 CONTINUE	driv4360
C		driv4370
	IF (IRPT.EQ. 3) THEN	driv4390
	IF(IPARM.EQ. 1) CALL SUBSOL (PARM3,PARM4,GMT,IDAY)	driv4400
	GO TO 555	driv4410
	END IF	driv4420
C		driv4430
C	C*****CARD 4 WAVENUMBER	driv4440
C		driv4450
	400 CONTINUE	driv4460
	IF (.NOT. SEA) THEN	
	READ(IRD,'(5I10)')IV1,IV2,IDV,IFWHM,IFILTER	
	END IF	
401	WRITE(IPR,'(15H0 CARD 4 *****5I10)')IV1,IV2,IDV,IFWHM,IFILTER	
	IF(IDV . LE. 0)THEN	driv4530
	PRINT *, ' ERROR IN IDV ',IDV	driv4540
	IDV = 1	driv4550
	ENDIF	driv4560



```

IF(IFWHM . LE. 0)THEN
    PRINT *, ' ERROR IN IFWHM ',IFWHM
    IFWHM = 2
ENDIF
IF ((IFILTER .GE. 1) .AND. (IFILTER .LE. 6)) THEN
C      reset wavenumbers to span filter passband:
        W1 = FLIST(1, IFILTER)
        W2 = FLIST(2, IFILTER)
        IV1 = INT(1E4/W2) - IDV
        IV2 = INT(1E4/W1) + IDV
    ELSE
C      filter data are absent. Reset to no filter at all:
        IFILTER = 0
    END IF
    IF (SeaSwitch) THEN
C      Check number of wavenumber steps. Reset, if necessary, to
C      prevent sea arrays from overflowing in "TRANS".
        Nv = (IV2 - IV1)/IDV
        IF (Nv .GE. Kv) IDV = (IV2 - IV1)/Kv + 1
    END IF
    WRITE(IP4, '(1H\, T20, 22HOUTPUT FILE FOR FILTER, I2,
+           2H: , I5, 3H TO, I5, 9H CM-1 IN , I2,
+           12H CM-1 STEPS., /1H\)' ) IFILTER, IV1, IV2, IDV
    WRITE(IP4, '(1H\, T65, 18H FILTERED RADIANCE)' )
    WRITE(IP4, '(45H\           ELEV.      ANGLE RANGE   TRANS ,
+           T49, 34H PATH      SEA      SKY      SUN,
+           T88, 15H TOTAL    TEMP.)' )
    WRITE(IP4, '(45H\           (mrad)    (deg)    (km)    (--) ,
+           T68,13H (W m-2 sr-1), T100, 3H(C), /)' )
    IF(IHAZE.EQ.3) THEN
C      IF(V1.LT.250.0 .OR. V2.LT.250.0) THEN
        IF(IV1.LT.250)THEN
            IHAZE=4
            WRITE (IPR,1203)
        ENDIF
1203      FORMAT('0**WARNING** NAVY MODEL IS NOT USABLE BELOW 250CM-1',
1      /,10X,' PROGRAM WILL SWITCH TO IHAZE=4 LOWTRAN 5 MARITIME',//)
    END IF
    IF (IRPT.EQ.4) GO TO 550
cc  IF (IRPT.EQ.-4) GO TO 560
500  CONTINUE
    IF (IRPT.EQ.3) GO TO 555
    WRITE(IPR,1410) (HTRRAD(I1,IEMSCT+1),I1=1,6)
1410  FORMAT('0 PROGRAM WILL COMPUTE ',6A4)
    IF(ISOURC .EQ. 1) WRITE(IPR,1204)
1204  FORMAT(' LUNAR SOURCE ONLY ')
    IF (IMULT .EQ. 1) THEN
        IF(IEMSCT.EQ.0 .OR. IEMSCT.EQ.3 ) THEN
            WRITE(IPR,1411)
1411      FORMAT('0 MULTIPLE SCATTERING HAS BEEN TURNED OFF ')
            WRITE (IP6,
+            '(/, 39HMULTIPLE SCATTERING HAS BEEN TURNED OFF)')
            IMULT=0
        ELSE
            WRITE(IPR,1412)
        END IF
    END IF

```

driv4570  
driv4580  
driv4590  
driv4600

driv4610  
driv4620  
driv4630  
driv4640  
driv4650  
driv4660  
driv4670  
driv4680  
driv4690  
driv4700  
driv4710  
driv4720  
driv4730  
driv4740  
driv4750  
driv4760  
driv4770  
driv4780  
driv4790  
driv4800  
driv4810

driv4820  
driv4830  
driv4840  
driv4850  
driv4860



```

1412 FORMAT('0 CALCULATIONS WILL BE DONE USING MULTIPLE SCATTERING ') driv4870
MDEL=MODEL driv4880
IF(MDEL.EQ.0)MDEL=8 driv4890
MM1=MDEL driv4900
MM2=MDEL driv4910
MM3=MDEL driv4920
IF(M1.NE.0)MM1=M1 driv4930
IF(M2.NE.0)MM2=M2 driv4940
IF(M3.NE.0)MM3=M3 driv4950
IF(MODEL.EQ.0) GO TO 510 driv4960
WRITE(IPR,1500) MM1,(HMODEL(I1,MM1),I1=1,5),MM2,(HMODEL(I2,MM2), driv4970
1 I2=1,5),MM3,(HMODEL(I3,MM3),I3=1,5) driv4980
1500 FORMAT('0 ATMOSPHERIC MODEL',/, driv4990
1 10X,'TEMPERATURE = ',I4,5X,5A4,/, driv5000
1 10X,'WATER VAPOR = ',I4,5X,5A4,/, driv5010
1 10X,'OZONE = ',I4,5X,5A4) driv5020
WRITE(IPR,1501) M4,M5,M6,MDEF driv5030
1501 FORMAT(20X,' M4 = ',I5,' M5 = ',I5,' M6 = ',I5,' MDEF = ',I5) driv5040
C driv5050
510 IF(JPRT.EQ.0) GO TO 520 driv5060
IF(ISEASN.EQ.0)ISEASN=1 driv5070
IF(IVULCN.LE.0) IVULCN=1 driv5080
IHVUL=IVULCN+10 driv5090
IF( IVULCN .EQ. 6) IHVUL = 11 driv5100
IF( IVULCN .EQ. 7) IHVUL = 11 driv5110
IF( IVULCN .EQ. 8) IHVUL = 13 driv5120
IHMET=1 driv5130
IF(IVULCN.GT.1)IHMET=2 driv5140
IF(IHAZE.EQ.0) GO TO 520 driv5150
WRITE(IPR,1510) (HHAZE(I,IHAZE),I=1,5),VIS,(HHAZE(I2,6),I2=1,5), driv5160
1 (HHAZE(II,6),II=1,5),(HSEASN(IA,ISEASN),IA=1,5), driv5170
2 (HHAZE(I3,IHVUL),I3=1,5), driv5180
3 (HVULCN(IB,IVULCN),IB=1,5),(HSEASN(IC,ISEASN),IC=1,5), driv5190
4 (HHAZE(I4,16),I4=1,5),(HMET(I5,IHMET),I5=1,5) driv5200
1510 FORMAT('0 AEROSOL MODEL',/,10X,'REGIME', driv5210
A T35,'AEROSOL TYPE',T60,'PROFILE',T85,'SEASON',/,/, driv5220
B 10X,'BOUNDARY LAYER (0-2 KM)',T35,5A4,T60,F5.1, driv5230
C ' KM VIS AT SEA LEVEL',/,10X,'TROPOSPHERE (2-10KM)',T35, driv5240
D 5A4,T60,5A4,T85,5A4,/,10X,'STRATOSPHERE (10-30KM)', driv5250
E T35,5A4,T60,5A4,T85,5A4,/,10X,'UPPER ATMOS (30-100KM)', driv5260
F T35,5A4,T60,5A4) driv5270
520 CONTINUE driv5280
IF(ITYPE.EQ.1) THEN
WRITE(IPR,1515) H1,RANGE driv5290
WRITE(IP6,1515) H1,RANGE
END IF
1515 FORMAT(/,' HORIZONTAL PATH',/,/,
1 8X,'ALTITUDE = ',F10.3,' KM',/,/,
2 8X,'RANGE = ',F10.3,' KM',/,/) driv5310
IF(ITYPE.EQ.2) THEN
WRITE(IPR,1516) H1,H2,ANGLE,RANGE,BETA,LEN driv5320
WRITE(IP6,1516) H1,H2,ANGLE,RANGE,BETA,LEN
END IF
1516 FORMAT(/,' SLANT PATH, H1 TO H2',/,/,
1 10X,'H1 = ',F10.3,' KM',/,10X,'H2 = ',F10.3,' KM',/,/, driv5340
2 10X,'ANGLE = ',F10.3,' DEG',/,10X,'RANGE = ',F10.3,' KM',/,/, driv5350
3 10X,'BETA = ',F10.3,' DEG',/,10X,'LEN = ',I6,/,) driv5360

```



```

      IF(ITYPE.EQ.3) THEN
        WRITE(IPR,1517) H1,H2,ANGLE
        WRITE(IP6,1517) H1,H2,ANGLE
      END IF
1517 FORMAT(/,'SLANT PATH TO SPACE',/,
1 10X, 'H1      = ',F10.3,' KM',/,
2 10X, 'HMIN    = ',F10.3,' KM',/,
3 10X, 'ANGLE   = ',F10.3,' DEG',/)
      IF (IEMSCT.NE.2) GO TO 550
C
C*****INTREPRET SOLAR SCATTERING PARAMETERS
C
C      IF (IPARM.EQ.1) CALL SUBSOL (PARM3,PARM4,GMT,IDAY)
C
      WRITE (IPR,1530)
1530 FORMAT('0 SINGLE SCATTERING CONTROL PARAMETERS SUMMARY '/')
      IF(IPARM.NE.2) WRITE (IPR,1532) PARM1,PARM2,PARM3,PARM4,GMT,PSIPO
1, IDAY
1532 FORMAT(10X,'OBSERVER LATITUDE =',T35,F10.2,' DEG NORTH OF EQUATOR',
1 //,10X,'OBSERVER LONGITUDE=',T35,F10.2,' DEG WEST OF GREENWICH',
2 //,10X,'SUBSOLAR LATITUDE =',T35,F10.2,' NORTH OF EQUATOR',/,
3 10X,'SUBSOLAR LONGITUDE =',T35,F10.2,' WEST OF GREENWICH',/,
4 10X,'TIME (<0 IS UNDEF)=' ,T35,F10.3,' GREENWICH TIME',/,
5 10X,'PATH AZIMUTH =',T35,F10.3,' DEG EAST OF NORTH',/,
6 10X,'DAY OF YEAR =',T35,I10)
      IF (IPARM.EQ.2) WRITE (IPR,1534)PARM1,PARM2,GMT,PSIPO,IDAY
1534 FORMAT(10X,'RELATIVE AZIMUTH =',T35,F10.3,' DEG EAST OF NORTH',/,
1 10X,'SOLAR ZENITH =',T35,F10.3,' DEG ',/,
2 10X,'TIME (<0 UNDEF) =',T35,F10.3,' GREENWICH TIME',/,
3 10X,'PATH AZIMUTH =',T35,F10.3,' DEG EAST OF NORTH',/,
4 10X,'DAY OF THE YEAR =',T35,I6)
      IF (ISOURC.EQ.0) WRITE (IPR,1535)
1535 FORMAT('0 EXTRATERRESTIAL SOURCE IS THE SUN')
      IF (ISOURC.EQ.1) WRITE (IPR,1536) ANGLEM
1536 FORMAT('0 EXTRATERRESTIAL SOURCE IS THE MOON, MOON PHASE ANGLE =',
1 F10.2,' DEG')
      IF (IPH.EQ.0) WRITE (IPR,1538) G
1538 FORMAT('0 H-G PHASE FUNCTION ,G=',F10.3)
      IF (IPH.EQ.1) WRITE (IPR,1540)
1540 FORMAT('0 USER SUPPLIED PHASE FUNCTION')
      IF (IPH.EQ.2) WRITE (IPR,1542)
1542 FORMAT('0 PHASE FUNCTION FROM MIE DATA BASE')
550 CONTINUE
C      V1 =FLOAT(INT(V1/5.0+0.1))*5.0
C      V2 =FLOAT(INT(V2/5.0+0.1))*5.0
C      TO AVOID THE DIFFICULTY FOR V1=0
      ALAM1= 99999.98
C      IF(V1.GT.0.)ALAM1=10000./V1
C      ALAM2=10000./V2
C      IF(DV.LT.5.)DV=5.
C      DV=FLOAT(INT(DV/5+0.1))*5.0
C      WRITE (IPR,1555) V1,ALAM1,V2,ALAM2,DV
C1555 FORMAT('0 FREQUENCY RANGE '//,10X,' V1 = ',F12.1,' CM-1 (' ,
C 1 F10.2,' MICROMETERS)',/,10X,' V2 = ',F12.1,' CM-1 (' ,F10.2,
C 2 ' MICROMETERS)',/,10X,' DV = ',F12.1,' CM-1')
C      IF(.NOT.MODTRN)THEN

```



```

IV1=5*(IV1/5) driv5900
IV2=5*((IV2+4)/5) driv5910
IDV=5+5*((IDV-5)/5) driv5920
ENDIF driv5930
IF (IV2.LT.IV1+IDV) THEN driv5940
  WRITE(IPR,'(/41H IV2 WAS LESS THAN IV1 + IDV AND HAS BEEN, driv5950
1 6H RESET,/)'') driv5960
  IV2=IV1+IDV driv5970
ENDIF driv5980
CRZ IF (MODTRN) THEN driv5990
CRZ IV1SAV=IV1 driv6000
CRZ IV2SAV=IV2 driv6010
CRZ IDVSAV=IDV driv6020
CRZ ENDIF driv6030
IF (IV1.NE.0) ALAM1=10000./IV1 driv6040
ALAM2=10000./IV2 driv6050
IF (IFWHM.LT.1) IFWHM=1 driv6060
IF (IFWHM.GT.50) IFWHM=50 driv6070
WRITE(IPR,'(17H0 FREQUENCY RANGE,/10X,8H IV1 =,I10,8H CM-1 (, driv6080
1 F10.2,13H MICROMETERS),/10X,8H IV2 =,I10,8H CM-1 (,F10.2, driv6090
2 13H MICROMETERS),/10X,8H IDV =,I10,5H CM-1,/10X,8H IFWHM =, driv6100
3 I10,5H CM-1') IV1,ALAM1,IV2,ALAM2,IDV,IFWHM driv6110
WRITE(IP6,'(15HFREQUENCY RANGE,/10X,9HIV1 =,I11,8H CM-1 (,
1 F7.2,13H MICROMETERS),/10X,9HIV2 =,I11,8H CM-1 (,F7.2,
2 13H MICROMETERS),/10X,9HIDV =,I11,5H CM-1,/10X,9HIFWHM =,
3 I11,5H CM-1,/10X,9HIFILTER =,I11)')
4 IV1,ALAM1,IV2,ALAM2,IDV,IFWHM,IFILTER
C driv6120
C*****LOAD ATMOSPHERIC PROFILE INTO /MODEL/ driv6130
C driv6140
C CALL STDMDL driv6150
C driv6160
C DEFINE COUNTER ITEST TO PREVENT ZENITH ANGLE QTHETA AND LAYER driv6170
C PATH LENGTH PL FROM BEING CHANGED DURING SOLAR CALCULATIONS driv6180
555 DO 15 I=1,102 driv6190
  DO 15 J=1,KMAX driv6200
    WPATH(I,J)=0.0 driv6210
  15 WPATHS(I,J)=0.0 driv6220
C driv6230
C ITEST=0 driv6240
C driv6250
C IF (IMULT.EQ. 1) THEN driv6260
  H1=ZM(1) driv6270
  H2=ZM(ML) driv6280
  ITYPE = 2 driv6290
  ANGLE = 0. driv6300
  BETA = 0. driv6310
  RANGE = 0. driv6320
  ISSGS = ISSGEO driv6330
  ISSGEO = 0 driv6340
C CALL GEO (IERROR,BENDNG,MAXGEO) driv6350
  MSOFF=68 driv6360
  CALL GEO (IERROR,BENDNG,MAXGEO,MSOFF) driv6370
  W15SV = W(15) driv6380
C driv6390
C W15SV IS THE REL HUM FROM 0 TO SPACE driv6400
C THIS REL HUM MAY BE DIFFERENT THAN THE PATH REL HUM driv6410

```



```

C      WHEN REL HUM ARE DIFFERENT THE ANSWER CAN CHANGE      driv6420
C
C      ISSGEO = ISSGS      driv6430
C      IMSMX=IKMAX      driv6440
C      DO 35 N=1,IMSMX      driv6450
C          PLST(N)=PL(N)      driv6460
C          DO 35 K=1,KMAX      driv6470
C35      WPMS(N,K)=WPATH(N,K)      driv6480
C          35      PLST(N)=PL(N)      driv6490
C      IF(IEMSCT.EQ.2) THEN      driv6500
C          CALL SSSEO(IERROR,IPH,IPARM,PARM1,PARM2,      driv6510
C              PARM3,PARM4,PSIPO,G,MAXGEO)      driv6520
C              PARM3,PARM4,PSIPO,G,MAXGEO,MSOFF)      driv6530
C          DO 30 N=1,IKMAX      driv6540
C              CSENSV(N) = ABS(CSZEN(N))      driv6550
C              IF(CSENSV(N) .LT. 0.0174) CSENSV(N) = 0.0174      driv6560
C          30      CONTINUE      driv6570
C          DO 45 N=1,ML      driv6580
C              DO 45 K=1,KMAX      driv6590
C                  WPMSS(N,K)=WPATHS(N,K)      driv6600
C          45      CONTINUE      driv6610
C      ENDIF      driv6620
C      H1 = H1S      driv6630
C      H2 = H2S      driv6640
C      ANGLE = ANGLES      driv6650
C      RANGE = RANGS      driv6660
C      BETA = BETAS      driv6670
C      ITYPE = ITYPES      driv6680
C      LEN = LENS      driv6690
C*****TRACE PATH THROUGH THE ATMOSPHERE AND CALCULATE ABSORBER AMOUNTS      driv6700
C      ISSGEO=0      driv6710
C      MSOFF=0      driv6720
C      ***** Save original value of SEA (false if earth not yet hit) *****      driv6730
C      SEAold = SEA      driv6740
C      CALL GEO(IERROR,BENDNG,MAXGEO,MSOFF)      driv6750
C      ***** and set HIT true if the earth has been hit within FNDHMN: *****      driv6760
C      HIT = ((.NOT. SEAold) .AND. SEA)
C      IF (HIT) THEN
C          WRITE (IP6, '(/, 6HSEA AT, F7.2,
+      31H K REPLACES BLACK BODY BOUNDARY,/,10X,9HUPWIND =,F11.3,
+      26H DEG EAST OF LINE OF SIGHT)') TBOUNDold, Psi
C          calculate geometry from point of view of the footprint
C          IF (IEMSCTold.EQ. 1) Pr = Psi*d2r + pi
C          IF ((IPARM.EQ. 0) .OR. (IPARM.EQ. 1)) THEN
C              ThetaO = PARM1
C              PhiO = PARM2
C              ThetaS = PARM3
C              PhiS = PARM4
C              CALL Foot(ThetaO,PhiO,ThetaS,PhiS,PsiPO,Beta,Psi)
C          ELSE IF (IPARM.EQ. 2) THEN
C              PsiO = PARM1
C              DelO = PARM2

```



	CALL SunFoot(Psi0,Del0,PsiPO,Beta,Psi)	
	END IF	
C	and issue new sky (and sun) cards in file 'TAPE5.SEA'	
	CALL Card	
	END IF	
	IF ((SeaSwitch) .AND. (.NOT. Sea)) THEN	
	WRITE (IP6, '(/, 13HTBOUND SET TO, F7.2,	
+	17H K FOR MARINE SKY)') TBOUND	
	END IF	
C	CALL AERTMP	driv6800
	IF(IMULT. NE. 1) W15SV = W(15)	driv6810
C		driv6820
C	SAVE TEMPERATURE AND PATH INFO FOR LATER USE	driv6830
C		driv6840
	IF(IMULT .EQ. 1) THEN	driv6850
	DO 25 N=1,IKMAX	driv6860
25	QTHETS(N) = QTHETA(N)	driv6870
	ENDIF	driv6880
C		driv6890
	IF(IERROR.GT.0) GO TO 630	driv6900
	IF(IEMSCT.EQ.3 .AND. IERROR.EQ. -5) GO TO 557	driv6910
	GO TO 558	driv6920
557	CONTINUE	driv6930
	WRITE(IPR,1557)	driv6940
1557	FORMAT('0 DIRECT PATH TO SUN INTERSECTS THE EARTH: SKIP TO ',	driv6950
1	'NEXT CASE')	driv6960
	GO TO 630	driv6970
558	CONTINUE	driv6980
C		driv6990
	IF(IEMSCT.EQ.2)CALL SSCEO(IERROR,IPH,IPARM,PARM1,PARM2,PARM3,	driv7000
C	1 PARM4,PSIPO,G,MAXGEO)	driv7010
1	PARM4,PSIPO,G,MAXGEO,MSOFF)	driv7020
	W(15) = W15SV	driv7030
C		driv7040
C	W15SV IS THE REL HUM (FOR MULT SCAT THIS MAY BE DIFFERENT	driv7050
C	FROM PATH REL HUM)	driv7060
C		driv7070
C	THE SECOND CALL TO SSCEO IS TO GET THE CORRECT ANGLES FOR	driv7080
C	PHASE FUNCTIONS	driv7090
C		driv7100
C	SAVE SOLAR PATH INFORMATION	driv7110
C		driv7120
	IF(IERROR.GT.0) GO TO 630	driv7130
C		driv7140
	IF(IMULT.EQ.1) THEN	driv7150
	DO 60 IK = 1,IMSMX	driv7160
	PL(IK)=PLST(IK)	driv7180
	IF(IEMSCT.EQ.2) CSZEN(IK)=CSENSV(IK)	driv7190
60	CONTINUE	driv7200
	DO 70 IK = 1,IKMAX	
70	QTHETA(IK) = QTHETS(IK)	driv7170
	ENDIF	driv7210
C		driv7220
C	C*****LOAD AEROSOL EXTINCTION, ABSORPTION, AND ASYMMETRY COEFFICIENTS	driv7230
C		driv7240
	CALL EXABIN	driv7250



C		driv7260
C*****	WRITE HEADER DATA TO TAPE 7	driv7270
C		driv7280
C560	WRITE(IPU,1110)MODEL, ITYPE, IEMSCT, IMULT, M1, M2, M3,	driv7290
C	1 M4, M5, M6, MDEF, IM, NOPRT, TBOUND, SALB	driv7300
560	WRITE(IPU, ' (L1, I4, 12I5, F8.3, F7.2) ') MODTRN, MODEL	driv7310
1	, ITYPE, IEMSCT, IMULT, M1, M2, M3, M4, M5, M6, MDEF, IM, NOPRT, TBOUND, SALB	driv7320
C	WRITE(IPR1,1110)MODEL, ITYPE, IEMSCT, IMULT, M1, M2, M3,	driv7330
C	1 M4, M5, M6, MDEF, IM, NOPRT, TBOUND, SALB	driv7340
	WRITE(IPR1, ' (L1, I4, 12I5, F8.3, F7.2) ') MODTRN, MODEL	driv7350
1	, ITYPE, IEMSCT, IMULT, M1, M2, M3, M4, M5, M6, MDEF, IM, NOPRT, TBOUND, SALB	driv7360
	WRITE(IPU,1200) IHAZE, ISEASN, IVULCN, ICSTL, ICLD, IVSA, VIS, WSS, WHH,	driv7370
1	RAINRT, GNDALT	driv7380
	WRITE(IPR1,1200) IHAZE, ISEASN, IVULCN, ICSTL, ICLD, IVSA, VIS, WSS, WHH,	driv7390
1	RAINRT, GNDALT	driv7400
	WRITE(IPU,1210) CTHIK, CALT, CEXT, ISEED	driv7410
	WRITE(IPR1,1210) CTHIK, CALT, CEXT, ISEED	driv7420
	WRITE(IPU,1230) ZCVSA, ZTVSA, ZINVS	driv7430
	WRITE(IPR1,1230) ZCVSA, ZTVSA, ZINVS	driv7440
	WRITE(IPU,1255) ML, (HMODEL(I,7), I=1,5)	driv7450
	WRITE(IPR1,1255) ML, (HMODEL(I,7), I=1,5)	driv7460
1255	FORMAT( I5,18A4)	driv7470
	IF(MODEL.NE.0)WRITE (IPU,1312) H1, H2, ANGLE, RANGE, BETA, RO, LEN	driv7480
	IF(MODEL.NE.0)WRITE (IPR1,1312) H1, H2, ANGLE, RANGE, BETA, RO, LEN	driv7490
	HMDLZ(8) = RANGE	driv7500
	IF(MODEL.EQ.0) WRITE(IPU,1560) (HMDLZ(K), K=1,8)	driv7510
	IF(MODEL.EQ.0) WRITE(IPR1,1560) (HMDLZ(K), K=1,8)	driv7520
1560	FORMAT(3F10.3, 5E10.3)	driv7530
	WRITE(IPU,1320) IPARM, IPH, IDAY, ISOURC	driv7540
	WRITE(IPR1,1320) IPARM, IPH, IDAY, ISOURC	driv7550
	WRITE(IPU,1322) PARM1, PARM2, PARM3, PARM4, GMT, PSIPO, ANGLEM, G	
	WRITE(IPR1,1322) PARM1, PARM2, PARM3, PARM4, GMT, PSIPO, ANGLEM, G	
3	WRITE(IPU,1400) V1, V2, DV	driv7580
3	WRITE(IPR1,1400) V1, V2, DV	driv7590
	WRITE(IPU, ' (5I10) ') IV1, IV2, IDV, IFWHM, IFILTER	
	WRITE(IPR1, ' (5I10) ') IV1, IV2, IDV, IFWHM, IFILTER	driv7620
C		driv7630
CRZ	IRAIN=0	driv7640
CRZ	IF(RAINRT.GT.0) IRAIN=1	driv7650
CCC		driv7660
CCC	CALCULATE EQUIVALENT LIQUID WATER CONSTANTS	driv7670
CCC		driv7680
	CALL EQUWLC	
	IF (SEA) THEN	
	Msea = Msea + 1	
	Done = (((IEMSCTold .EQ. 1) .AND. (Msea .EQ. 3))	
+	.OR. ((IEMSCTold .EQ. 2) .AND. (Msea .EQ. 4)))	
	IF (Done) THEN	
	READ(IRD, 1600) IRPT	
	ELSE	
	READ(IRDS,1600) IRPT	
	END IF	
	ELSE	
	READ(IRD, 1600) IRPT	
	END IF	
1600	FORMAT(I5)	driv7720
	WRITE(IPU,1600) IRPT	driv7730

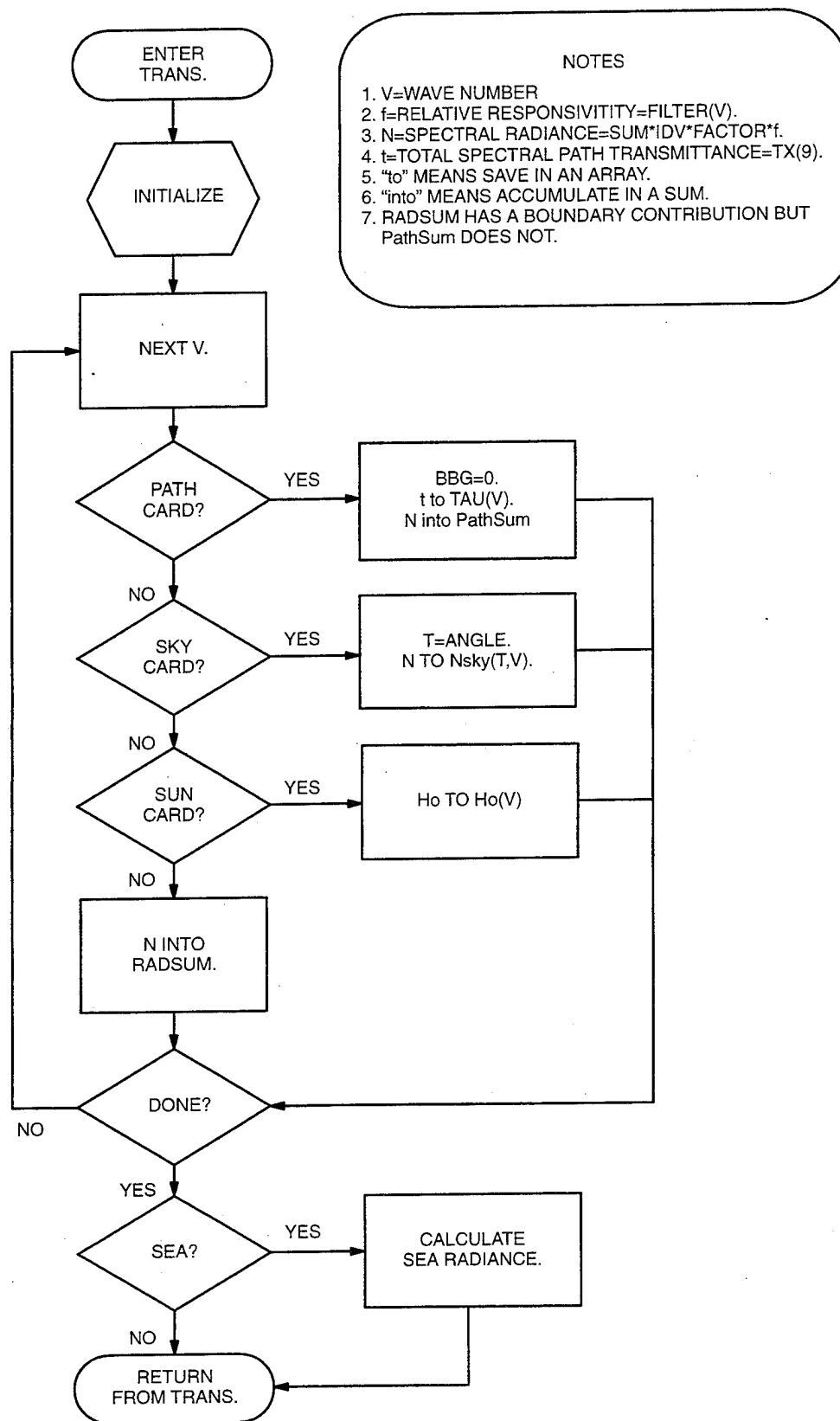






**APPENDIX C**  
**MODIFIED SUBROUTINE "TRANS"**





**Figure C-1.** Flowchart for modified subroutine "TRANS."



	SUBROUTINE TRANS(IPH,ISOURC,IDAY,ANGLEM,ground)	
C		tras 110
C	CALCULATES TRANSMITTANCE AND RADIANCE VALUES BETWEEN IV1 AND IV2	tras 120
C	FOR A GIVEN ATMOSPHERIC SLANT PATH	tras 130
	parameter(nbins=99,iprint=50,maxv=50000)	tras 140
	real WGT(nbins),SLIT(56,nbins)	tras 150
	LOGICAL IVTEST,loop0,ground,transm,modtrn	tras 160
	COMMON RELHUM(34),WHNO3(34),ICH(4),VH(17),TX(63),W(63),IMSMX,	tras 170
1	WPATH(102,63),TBBY(102),PATM(102),NSPEC,KPOINT(12),	tras 180
2	ABSC(5,47),EXTC(5,47),ASYM(5,47),VX0(47),AWCCON(5)	tras 190
	COMMON /IFIL/ IRD,IPR,IPU,NPR,IPR1,IP6,IP7,IP8,IP4,IRDS,IP6S,	
+	ITR,Isky,Isun,Ipath	
	COMMON/CARD1/MODEL,ITYPE,IEMSCT,M1,M2,M3,IM,NOPRNT,TBOUND,SALB,	tras 210
1	MODTRN	tras 220
	COMMON /CARD2/ IHAZE,ISEASN,IVULCN,ICSTL,ICIR,IVSA,VIS,WSS,WHH,	
1	RAINRT	
	COMMON/CARD3/H1,H2,ANGLE,RANGE,BETA,REE,LEN	
	COMMON/CARD4/IV1,IV2,IDV,IFWHM,IFILTER	
	COMMON/CNSTNS/PIX,CA,DEG,GCAIR,BIGNUM,BIGEXP	
	COMMON/CNTRL/KMAX,M,IKMAX,NL,ML,IKLO,ISSGEO,IMULT	tras 250
	COMMON/SOLS/AH1(68),ARH(68),	tras 260
1	WPATHS(102,63),PA(68),PRX(68),ATHETA(35),ADBETA(35),LJ(69),	
2	JTURN,ANGSUN,CSZEN(68),TBBYS(102,12),PATMS(102,12)	tras 280
	COMMON/SRAD/TEB1,TEB2SV	tras 290
	COMMON/MSRD/TLE(34),COSBAR(34),OMEGA0(68),UPF(10,34),DNF(10,34),	tras 300
1	TAER(34),ASYIK(68),ASYDM(68),STRN(0:34),DMOLS(68),DSTRN(0:68),	tras 310
2	FDNSRT,FDNTRT,TAUT(34),UMF(34),DMF(34),UMFS(34),DMFS(34)	tras 320
	COMMON/ICLL/ICALL,FPHS,FALB,FORBIT	tras 330
	PARAMETER (Kr = 216, Kv = 400)	
	COMMON /Filters/ FLIST(5,6),	
+	FILTER1(45), BB1(Kr), FILTER2(54), BB2(Kr),	
+	FILTER3(39), BB3(Kr), FILTER4(47), BB4(Kr),	
+	FILTER5(101),BB5(Kr), FILTER6(75), BB6(Kr)	
	COMMON/Sea/ Sea,Hit,Msea,TBOUNDold,IEMSCTold	
	COMMON/Geometry/ To,Po,Tr,Pr	
	COMMON/Constants/pi,r2d,d2r,epsilon,delta,onem,onep,infinity	
	DIMENSION Tau(Kv), SkyN(3,Kv), Ho(Kv), Tsky(3), Rsky(3)	
	LOGICAL Sea, PathCard, SkyCard, LastSky, SunCard, Hit	
	REAL Npath, Nsky, Nvsky, Nsea, Nsun, Ntotal, Nbb, infinity, No	
	T0 = 273.15	
	common /solar/lsame	tras 340
	logical lsame	tras 350
C		tras 360
C	Initialize slit function array	tras 370
	DO 10 I = 1,56	tras 380
	DO 10 J = 1,nbins	tras 390
10	SLIT(I,J) =0.	tras 400
C		tras 410
C	Initialize radiance minimum and maximum parameters	tras 420
	RADMIN=bignum	tras 430
	RADMAX=0.	tras 440
C		tras 450
C	Initialize ground emissivity (one minus ground albedo)	tras 460
	EMISS=1.-SALB	tras 470
C		tras 480
C	Store the number of path layers in ikmx	tras 490
	IKMX=IKMAX	tras 500



C		tras 510
C	Initialize integrated absorption, radiance, solar irradiance and	tras 520
C	transmitted solar irradiance sums	tras 530
	SUMA=0.	tras 540
	RADSUM=0.	tras 550
	SSOL=0.	tras 560
	STSOL=0.	tras 570
	PathSum = 0.	
	PathCard = .FALSE.	
	SkyCard = .FALSE.	
	LastSky = .FALSE.	
	SunCard = .FALSE.	
	IF (Sea) THEN	
	IF (Msea .EQ. 0) THEN	
	PathCard = .TRUE.	
	ELSE IF ((Msea .GE. 1) .AND. (Msea .LE. 3)) THEN	
	SkyCard = .TRUE.	
	ELSE IF (Msea .EQ. 4) THEN	
	SunCard = .TRUE.	
	END IF	
	IF (Msea .EQ. 3) THEN	
	LastSky = .TRUE.	
	END IF	
	END IF	
	IF (SkyCard) Tsky(Msea) = ANGLE*d2r	
	Istore = 0	
C		tras 580
C	Initialize integration weighting factor	tras 590
	FACTOR=.5	tras 600
C		tras 610
C	Initialize icount, used to determine when header must be printed	tras 620
	ICOUNT=iprint	tras 630
C		tras 640
C	Do not perform a MODTRAN calculation if all sources are continuum	tras 650
CSSI	IF (IV1.GE.22655) modtrn=.false.	tras 660
	IF (IV1.GE.22681) modtrn=.false.	tras 670
	IF (modtrn) THEN	tras 680
		tras 690
C	WHEN THE band model or line-by-line option is used, call	tras 700
C	routine "bmdata" to INITIALIZE PARAMETERS AND TO SET THE	tras 710
C	FREQUENCY STEP SIZE "IDVX" TO THE BAND WIDTH (1 CM-1).	tras 720
	IDV5=5	tras 730
	CALL bmdata(IV1,IFWHM,IDVX,IKMX,MXFREQ)	tras 740
	IWIDM1=IFWHM/IDVX-1	tras 750
	IV=5*((IV1-IWIDM1)/5)	tras 760
	IF (IV.LT.0) IV=0	tras 770
	IVX=IV-IDVX	tras 780
	IV=IV-5	tras 790
	IVXMAX=IV2+IWIDM1	tras 800
	ELSE	tras 810
	IDV5=IDV	tras 820
	IDVX=IDV5	tras 830
	IWIDM1=0	tras 840
	IV=IV1-IDV5	tras 850
	IVX=IV	tras 860
	IVXMAX=IV2+IWIDM1	tras 870
	IF (IVXMAX.GT.maxv) IVXMAX=maxv	tras 880



```

      IF (IDV.LT.5) IDV=5
      ENDIF
      IWRITE=IV1+IWIDM1
      IWIDTH=IWIDM1+1
C
C   PERFORM TRIANGULAR SLIT INITIALIZATION. TRANSMITTANCES AT A
C   GIVEN FREQUENCY CONTRIBUTE TO 2*IWIDTH-1 TRIANGULAR SLITS.
C   THESE CONTRIBUTIONS ARE STORED IN ARRAY SLIT. WGT IS THE
C   NORMALIZED WEIGHT USED TO DEFINE THE TRIANGLE.
      NWGT=2*IWIDTH
      WNORM=1./((IWIDTH*IWIDTH))
      DO 20 I=1,IWIDTH
        WGT(I)=I*WNORM
20    WGT(NWGT-I)=wgt(i)
      NWGT=NWGT-1
      NWGTM1=NWGT-1
C
C   Initialize ICALL (= 0 for initial call to subroutine source)
      ICALL=0
C
C   Initialize transm (.true. for transmittance only calculations)
      transm=.true.
      IF (IEMSCT.EQ.1 .OR. IEMSCT.EQ.2) transm=.false.
C
C   Print headers
      IF (IEMSCT.EQ.0) THEN
        WRITE (IPU, '(46H \FREQ  TOTAL  H2O  CO2+ OZONE  TRACE,
1      49H  N2 CON  H2O CON MOL SCAT AER-HYD HNO3  AER-HYD)')
        WRITE (IP7, '(46H \FREQ  TOTAL  H2O  CO2+ OZONE  TRACE,
1      49H  N2 CON  H2O CON MOL SCAT AER-HYD HNO3  AER-HYD)')
        WRITE (IPR1, '(45H \FREQ  H2O  O3  CO2  CO  CH4,
1      47H  N2O  O2  NH3  NO  NO2  SO2,/
2      55H  \1/CM TRANS  TRANS  TRANS  TRANS  TRANS  TRANS,
3      39H  TRANS  TRANS  TRANS  TRANS  TRANS)')
        WRITE (IP8, '(45H \FREQ  H2O  O3  CO2  CO  CH4,
1      47H  N2O  O2  NH3  NO  NO2  SO2,/
2      55H  \1/CM TRANS  TRANS  TRANS  TRANS  TRANS  TRANS,
3      39H  TRANS  TRANS  TRANS  TRANS  TRANS)')
      ELSE IF (IEMSCT.EQ.3) THEN
        WRITE (IPU, '(32H \FREQ  TRANS  SOL TR  SOLAR)')
        WRITE (IP7, '(2H \, T25,
+      40HIRRADIANCE (W M-2) PASSED THROUGH FILTER,
+      I2)') IFILTER
        WRITE (IP7, '(32H \FREQ  TRANS  SOL TR  SOLAR)')
      ELSE IF (IEMSCT.EQ.1) THEN
        WRITE (IPU, '(45H \FREQ  TRANS  ATMOS. RAD., T88,
+      18H- LOG TOTAL TRANS.)')
        WRITE (IP7, '(2H \, T25,
+      43HRADIANCE (W M-2 SR-1) PASSED THROUGH FILTER,
+      I2)') IFILTER
        WRITE (IP7, '(30H \FREQ  TRANS  ATMOS. RAD., T88,
+      18H- LOG TOTAL TRANS.)')
      ELSE IF (IEMSCT.EQ.2) THEN
        WRITE (IPU, '(42H \FREQ  TRANS  ATMOS  PATH  SINGLE,
1      28H  GROUND DIRECT  TOTAL RAD)')
        WRITE (IP7, '(2H \, T25,
+      43HRADIANCE (W M-2 SR-1) PASSED THROUGH FILTER,

```

```

      tras 890
      tras 900
      tras 910
      tras 920
      tras 930
      tras 940
      tras 950
      tras 960
      tras 970
      tras 980
      tras 990
      tras1000
      tras1010
      tras1020
      tras1030
      tras1040
      tras1050
      tras1060
      tras1070
      tras1080
      tras1090
      tras1100
      tras1110
      tras1120
      tras1130
      tras1140
      tras1150
      tras1160
      tras1170
      tras1180
      tras1190
      tras1210
      tras1220
      tras1230
      tras1240
      tras1250

```



```

+          I2)') IFILTER
+      WRITE(IP7,'(42H \FREQ   TRANS   ATMOS   PATH   SINGLE,
1      28H   GROUND DIRECT   TOTAL RAD, T88,
+      18H- LOG TOTAL TRANS.)')
+      END IF
+      If (PathCard) then
+          WRITE(IPath,'(1H\, T25,
+          52HRADIANCE (W M-2 SR-1 (CM-1)-1) PASSED THROUGH FILTER,
+          I2,/,1H\,/,40H\ V   T   f   Npath*f   INTEGRAL,
+          /,1H\') Ifilter
+      Else if (LastSky) then
+          WRITE(Isky,'(1H\, T25,
+          52HRADIANCE (W M-2 SR-1 (CM-1)-1) PASSED THROUGH FILTER,
+          I2,/,1H\,/,40H\ V   T   f   Nsky*T*f   INTEGRAL,
+          22H   Nsea*T*f   INTEGRAL,/,1H\') Ifilter
+      Else if (SunCard) then
+          WRITE(Isun,'(1H\, T25,
+          52HRADIANCE (W M-2 SR-1 (CM-1)-1) PASSED THROUGH FILTER,
+          I2,/,1H\,/,40H\ V   T   f   Nsun*T*f   INTEGRAL,
+          /,1H\') Ifilter
+      End If
+      IF(NOPRNT.EQ.-1)THEN
+          IF(IMULT.EQ.1)THEN
+              WRITE(IPR1,'(37H   \V   ALT1   UFLX   UFLXS,
1              50H   DFLX   DFLXS   DIRS   TRANS)')tras1270
+              WRITE(IP8, '(37H   \V   ALT1   UFLX   UFLXS,
1              50H   DFLX   DFLXS   DIRS   TRANS)')tras1280
+          ELSE
+              IF(IEMSCT.GT.0)WRITE(IPR1,'(23H   \V   ALT1   ALT2,
1              30H   B(V,T)   DTAU   TAU)')tras1290
+              WRITE(IP8, '(23H   \V   ALT1   ALT2,
1              30H   B(V,T)   DTAU   TAU)')tras1300
+          ENDIF
+      ENDIF
+      Initialize layer loop variables
+      loop0=.true.
+      call loop(loop0,iv,ivx,ikmx,mxfreq,summs,transm,iph,
1      sumssr,ivtest,unif,trace,sumv,isourc,iday,anglem,frac)
+      loop0=.false.
+      END INITIALIZATION, BEGIN OF FREQUENCY LOOP
+      "IVX" IS THE FREQUENCY AT WHICH TRANSMITTANCE WILL BE CALCULATED.
+      DURING THE FIRST PASS, "IVX" AND "IV" MUST BE EQUAL.
30 IVX=IVX+IDVX
+      IF(IV.LT.IVX)THEN
+          IV=IV+IDV5
+          IVTEST=.TRUE.
+      ELSE
+          IVTEST=.FALSE.
+      ENDIF
+      SET INTERPOLATION FRACTION.
+      FRAC=FLOAT(IV-IVX)/IDV5
+      IF(ICOUNT.EQ.iprint)THEN

```



```

c      Reinitialize counter and print header      tras1590
      ICOUNT=0      tras1600
      IF(IEMSCT.EQ.0) THEN      tras1610
        WRITE(IPR, '(1H1,/33H  FREQ WAVELENGTH TOTAL  H2O, tras1620
1          47H  CO2+  OZONE  TRACE  N2 CONT  H2O CONT, tras1630
2          47H MOL SCAT  AER-HYD  HNO3  AER-HYD  INTEGRATED, tras1640
3          //43H  1/CM  MICRONS  TRANS  TRANS  TRANS, tras1650
4          44H  TRANS  TRANS  TRANS  TRANS  TRANS, tras1660
5          40H  TRANS  TRANS  ABS  ABSORPTION,/)') tras1670
      ELSEIF(IEMSCT.EQ.1) THEN      tras1680
        WRITE(IPR, '(1H1,20X,28HRADIANCE(WATTS/CM2-STER-XXX), tras1690
1          /8H0  FREQ,T10,6HWAVLEN,T19,14HATMOS RADIANCE, tras1700
2          T39,9H INTEGRAL,T49,5HTOTAL,/2X,6H(CM-1), tras1710
3          T10,7H(MICRN),T19,6H(CM-1),T29,7H(MICRN), tras1720
4          T39,6H(CM-1),T49,5HTRANS,/)') tras1730
      ELSEIF(IEMSCT.EQ.3) THEN      tras1740
        WRITE(IPR, '(1H1,22X,27HIRRADIANCE (WATTS/CM2-XXXX), tras1750
1          /7H0  FREQ,T11,6HWAVLEN,T23,11HTRANSMITTED, tras1760
2          T45,5HSOLAR,T61,10HINTEGRATED,T80,5HTOTAL, tras1770
3          /2X,6H(CM-1),T10,7H(MICRN),T20,6H(CM-1), tras1780
4          T30,7H(MICRN),T40,6H(CM-1),T50,7H(MICRN), tras1790
5          T60,6HTRANS.,T70,5HSOLAR,T80,5HTRANS)') tras1800
      ELSEIF(IMULT.EQ.0) THEN      tras1810
        WRITE(IPR, '(1H1,45X,28HRADIANCE(WATTS/CM2-STER-XXX), tras1820
1          /7H0  FREQ,T11,6HWAVLEN,T21,14HATMOS RADIANCE, tras1830
2          T41,14HPATH SCATTERED,T61,16HGROUND REFLECTED, tras1840
3          T85,5HTOTAL,T99,8HINTEGRAL,T110,5HTOTAL, tras1850
4          /2X,6H(CM-1),T10,7H(MICRN),T20,6H(CM-1), tras1860
5          T30,7H(MICRN),T40,6H(CM-1),T50,7H(MICRN), tras1870
6          T60,6H(CM-1),T70,7H(MICRN),T80,6H(CM-1), tras1880
7          T90,7H(MICRN),T100,6H(CM-1),T110,5HTRANS,/)') tras1890
      ELSE      tras1900
        WRITE(IPR, '(1H1,45X,28HRADIANCE(WATTS/CM2-STER-XXX), tras1910
1          //6H0  FREQ,T10,6HWAVLEN,T20,14HATMOS RADIANCE,T40, tras1920
2          4HPATH,19H SCATTERED RADIANCE,T69, tras1930
3          25HGROUND REFLECTED RADIANCE,T100,14HTOTAL RADIANCE, tras1940
4          T118,8HINTEGRAL,T127,5HTOTAL,/T45,5HTOFL,T59, tras1950
5          6HS SCAT,T75,5HTOTAL,T89,6HDIRECT,/1X,6H(CM-1),T9, tras1960
6          7H(MICRN),T19,6H(CM-1),T29,7H(MICRN),T39,6H(CM-1), tras1970
7          T49,7H(MICRN),T59,6H(CM-1),T69,6H(CM-1),T79, tras1980
8          7H(MICRN),T89,6H(CM-1),T99,6H(CM-1),T109,7H(MICRN), tras1990
9          T119,6H(CM-1),T127,5HTRANS,/)') tras2000
      ENDIF      tras2010
    ENDIF      tras2020
c      Determine layer loop maximum      tras2030
c      IF(transm) THEN      tras2040
c      tras2050
c      tras2060
c      For transmission calculations, skip over layer loop in tratra tras2070
c      IKMAX=1      tras2080
c      ELSEIF(IMULT.EQ.1 .and. .not. lsame) THEN      tras2090
c      tras2100
c      FOR MULTIPLE SCATTERING SET IKMAX TO IMSMX      tras2110
c      IKMAX=IMSMX      tras2120
c      ELSE      tras2130
c      tras2140
c      IF NOT MULTIPLE SCATTERING, RESET IKMAX TO ORIGINAL VALUE tras2150

```



	IKMAX=IKMX	tras2160
	ENDIF	tras2170
	SUMV=0.	tras2180
C		tras2190
C	Initialize transmission array	tras2200
	TX(1)=1.	tras2210
	TX(2)=1.	tras2220
	TX(3)=1.	tras2230
	DO 40 K=4,KMAX	tras2240
40	TX(K)=0.	tras2250
	call loop(loop0,iv,ivx,ikmx,mxfreq,summs,transm,	tras2260
1	iph,sumssr,ivtest,unif,trace,sumv,isourc,iday,anglem,frac)	tras2270
C		tras2280
C	THE PARAMETERS "UNIF", "TRACE", "SUMV", "SUMSSR", "SUMMS"	tras2290
C	AND "TEB1" ARE TEMPORARILY STORED IN "TX" SO THAT THEIR	tras2300
C	CONVOLUTION OVER THE TRIANGULAR SLIT CAN BE CALCULATED.	tras2310
	TX(2)=UNIF	tras2320
	TX(3)=TRACE	tras2330
	TX(8)=SUMV	tras2340
	TX(12)=SUMSSR	tras2350
	TX(13)=SUMMS	tras2360
	TX(14)=TEB1	tras2370
	DO 60 K=2,56	tras2380
	IP1=NWGT	tras2390
	DO 50 I=NWGTM1,1,-1	tras2400
	SLIT(K,IP1)=SLIT(K,I)+WGT(IP1)*tx(k)	tras2410
50	IP1=I	tras2420
	SLIT(K,1)=WGT(1)*tx(k)	tras2430
60	TX(K)=SLIT(K,NWGT)	tras2440
C		tras2450
C	CHECK IF VALUES ARE TO BE PRINTED	tras2460
	IF (IVX.LT.IWRITE) GOTO30	tras2470
	IWRITE=IWRITE+IDV	tras2480
	IF (IWRITE.GT.IVXMAX) FACTOR=.5	tras2490
	ICOUNT=ICOUNT+1	tras2500
C		tran2510
C	RENORMALIZE IF TRIANGULAR SLIT EXTENDS TO NEGATIVE FREQUENCIES	tras2520
	IF (IVX.LT.NWGTM1) THEN	tras2530
	store=1.-.5*(NWGTM1-IVX)*(NWGTM1-IVX+1)*WNORM	tras2540
	DO 70 K=2,56	tras2550
70	TX(K)=TX(K)/store	tras2560
	ENDIF	tras2570
	UNIF=TX(2)	tras2580
	TRACE=TX(3)	tras2590
	SUMV=TX(8)	tras2600
	SUMSSR=TX(12)	tras2610
	SUMMS=TX(13)	tras2620
	TEB1=TX(14)	tras2630
	V=FLOAT(IVX-IWIDM1)	tras2640
	ALAM=1.0E+04/(V+.000001)	tras2650
	Istore = Istore + 1	
	Width = IDV*FACTOR	
	f = Filter(V,Ifilter)	
	SUMA=SUMA+(1.0-TX(9))*f*Width	
	ALTX9=BIGNUM	tras2670
	IF (TX(9).GT.0.) ALTX9=-LOG(TX(9))	tras2680
	GOTO(80,90,90,100), IEMSCT+1	tras2690



C			tras2700
C		TRANSMITTANCE ONLY	tras2710
	80	TX(10)=1.-TX(10)	tras2720
		TX(7)=TX(7)*TX(16)	tras2730
		WRITE(IPR, '(F8.0,F8.3,11F9.4,F12.3)')V,ALAM, TX(9), TX(17),	tras2740
	1	UNIF, TX(31), TRACE, TX(4), TX(5), TX(6), TX(7), TX(11), TX(10), SUMA	tras2750
		WRITE(IPR1, '(F7.0,11F8.4,1PE10.3)')V, TX(17), TX(31), TX(36),	tras2760
	1	TX(44), TX(46), TX(47), TX(50), TX(52), TX(54), TX(55), TX(56)	tras2770
		WRITE(IP8, '(F7.0,11F8.4,1PE10.3)')V, TX(17), TX(31), TX(36),	
	1	TX(44), TX(46), TX(47), TX(50), TX(52), TX(54), TX(55), TX(56)	
		WRITE(IPU, '(F7.0,11F8.4,1PE10.3)')V, TX(9), TX(17), UNIF,	tras2780
	1	TX(31), TRACE, TX(4), TX(5), TX(6), TX(7), TX(11), TX(10), ALTX9	tras2790
		WRITE(IP7, '(F7.0,11F8.4,1PE10.3)')V, TX(9), TX(17), UNIF,	
	1	TX(31), TRACE, TX(4), TX(5), TX(6), TX(7), TX(11), TX(10), ALTX9	
		GOTO110	tras2800
C			tras2810
C		ATMOSPHERIC RADIANCE INCLUDING EMISSION OF BOUNDARY	tras2820
C		ATTENUATED BY TOTAL TRANSMISSION	tras2830
C			tras2840
C		CALCULATE THERMAL RADIANCE CONTRIBUTION OF BOUNDARY AND	tras2850
C		ADD THE SCATTERED CONTRIBUTION TO THE THERMAL RADIANCE	tras2860
C		IF THE PATH INTERSECTS THE SURFACE	tras2870
	90	IF ((TBOUND.LE.0.) .OR. (PathCard)) THEN	
		BBG = 0.	tras2890
		ELSE	tras2900
		BBG=BBFN(TBOUND,V)*TX(9)*EMISS	tras2910
		IF(IMULT.EQ.1 .AND. ground) THEN	
		BBG=BBG+SALB*FDNTRT*TX(9)/PI	tras2920
		END IF	
		ENDIF	tras2930
C			tras2940
C		ADD THERMAL BOUNDARY AND MULTIPLE SCATTERED RADIANCE	tras2950
		SUMV=(SUMV+BBG)*f.	
		SUMVV=SUMV	tras2970
		IF (V.GT.0.) THEN	
		SUMV=(1.0E+08/V**2)*SUMV	! W m-2 sr-1 (cm-1)-1
		END IF	
		IF(IEMSCT.EQ.1) THEN	tras2990
		RADSUM=RADSUM + SUMV*Width	
		WRITE(IPR, '(F8.0,F8.3,1P3E10.2,0PF9.4)')	tras3010
	1	V, ALAM, SUMV, SUMVV, RADSUM, TX(9)	tras3020
		WRITE(IPU, '(F7.0,F8.4,1PE15.8,T96,E10.3)')	tras3030
	1	V, TX(9), SUMV, ALTX9	tras3040
		WRITE(IP7, '(F7.0,F8.4,1PE15.8,T96,E10.3)')	
	1	V, TX(9), SUMV, ALTX9	
		WRITE(IPR1, '(F7.0,11F8.4,1PE10.3)')V, TX(9), TX(17), UNIF,	tras3050
	1	TX(31), TRACE, TX(4), TX(5), TX(6), TX(7), TX(11), TX(10), ALTX9	tras3060
		WRITE(IP8, '(F7.0,11F8.4,1PE10.3)')V, TX(9), TX(17), UNIF,	
	1	TX(31), TRACE, TX(4), TX(5), TX(6), TX(7), TX(11), TX(10), ALTX9	
		SUMT=SUMV	tras3070
		SUMTT=SUMVV	tras3080
		ELSE	tras3090
C			tras3100
C		SOLAR SCATTERED RADIANCE	tras3110
		CALL SOURCE(V, ISOURC, IDAY, ANGLEM, SS)	tras3120
C			tras3130
C		MULTIPLY SUMSSR BY THE EXTRATERRESTRIAL SOURCE STRENGTH SStras3140	



```

SUMSSS=SUMSSR*SS                                     tras3150
C
C CALCULATE TOTAL SINGLE SCATTERED + MULTIPLE SCATTERED tras3160
C SOLAR RADIANCE FOR EACH FREQUENCY [W/CM2-STER-MICROMETER] tras3170
SUMSSR=SUMSSS+SUMMS                                     tras3180
store=0.                                                 tras3190
if(v.gt.0.)store=1.e8/v**2      ! [W m-2 sr-1 (cm-1)-1] tras3200
SUMS=store*SUMSSR                                     tras3220
SUMSSS=store*SUMSSS                                     tras3230
C
C RFLSOL IS GROUND-REFLECTED DIRECT SOURCE RADIANCE AND tras3240
RFLSOL=0.                                             tras3250
RFLS=0.                                             tras3260
RFLSS=0.                                             tras3270
RFLSSS=0.                                           tras3280
IF(ground .AND. TEB1.GT.0)THEN tras3290
  IF(ANGSUN.GE.0.)RFLSSS=SS*TEB1*SALB*COS(ANGSUN*CA)/PI tras3300
  RFLSOL=RFLSSS                                     tras3310
  IF(IMULT.EQ.1)RFLSOL=RFLSOL+SALB*FDNSRT*TX(9)/PI tras3320
  RFLS=STORE*RFLSOL                                 tras3330
  RFLSS=STORE*RFLSSS                                 tras3340
  RFLSSS=STORE*RFLSSS                                 tras3350
ENDIF                                                 tras3360
SUMT=SUMV+(SUMS+RFLS)*f
SUMTT=SUMVV+(SUMSSR+RFLSOL)*f
RADSUM=RADSUM + SUMT*Width
IF (IMULT.NE.1) THEN tras3400
  WRITE(IPR,'(F8.0,F8.3,1P9E10.2,0PF9.4)') tras3410
  V,ALAM,SUMV,SUMVV,SUMS,SUMSSR,RFLS,RFLSOL, tras3420
  SUMT,SUMTT,RADSUM,TX(9) tras3430
ELSE tras3440
  WRITE(IPR,'(F7.0,F8.3,1P11E10.2,0PF7.4)') tras3450
  V,ALAM,SUMV,SUMVV,SUMS,SUMSSR,SUMSSS,RFLS, tras3460
  RFLSOL,RFLSS,SUMT,SUMTT,RADSUM,TX(9) tras3470
END IF tras3480
WRITE(IPU,'(F7.0,F8.4,1P6E9.2,0P2F8.4,T96,1PE10.3)')V, tras3490
TX(9),SUMV,SUMS,SUMSSS,RFLS,RFLSS,SUMT,TEB1,TEB2SV,ALTX9tras3500
WRITE(IP7,'(F7.0,F8.4,1P6E9.2,0P2F8.4,T96,1PE10.3)')V,
TX(9),SUMV,SUMS,SUMSSS,RFLS,RFLSS,SUMT,TEB1,TEB2SV,ALTX9
WRITE(IPr1,'(F7.0,11F8.4,1PE10.3)')V,TX(9),TX(17),UNIF, tras3510
TX(31),TRACE,TX(4),TX(5),TX(6),TX(7),TX(11),TX(10),ALTX9 tras3520
WRITE(IP8,'(F7.0,11F8.4,1PE10.3)')V,TX(9),TX(17),UNIF,
TX(31),TRACE,TX(4),TX(5),TX(6),TX(7),TX(11),TX(10),ALTX9
END IF tras3530
IF (PathCard) THEN
  Tau(Istore) = TX(9)
  PathSum = PathSum + SUMT*Width
  Write(IPath,'(I5,2F7.3,2(1PE11.3))')
  V, TX(9), f, SUMT, PathSum
+
END IF
IF (SkyCard) SkyN(Msea,Istore) = SUMT
C
C GO TO 110 tras3540
C tras3550
C DIRECTLY TRANSMITTED SOLAR IRRADIANCE [WATTS/(CM2 MICROMETER)] tras3560
100 CALL SOURCE(V,ISOURC,IDAY,ANGLEM,SOLIL) tras3570
SOLIV=0. tras3580
IF(V.GT.0.)SOLIV=SOLIL*1.E+8/V**2      ! [W m-2 sr-1 (cm-1)-1]

```



```

TSOLIV=SOLIV*TX(9)*f
IF (SunCard) then
  Ho(Istore) = TSOLIV
END IF
TSOLIL=SOLIL*TX(9)*f
STSOL=STSOL+TSOLIV*Width
SSOL=SSOL+SOLIV*Width
WRITE(IPR, '(F8.0,F8.3,1P6E10.2,0PF9.4)')
1 V,ALAM,TSOLIV,TSOLIL,SOLIV,SOLIL,STSOL,SSOL,TX(9) tras3640
WRITE(IPU, '(F7.0,F8.4,1P2E9.2,T96,E10.3)') tras3650
1 V,TX(9),TSOLIV,SOLIV,ALTX9 tras3660
WRITE(IP7, '(F7.0,F8.4,1P2E9.2,T96,E10.3)') tras3670
1 V,TX(9),TSOLIV,SOLIV,ALTX9
WRITE(IPr1, '(F7.0,11F8.4,1PE10.3)') V,TX(9),TX(17),UNIF, tras3680
1 TX(31),TRACE,TX(4),TX(5),TX(6),TX(7),TX(11),TX(10),ALTX9 tras3690
WRITE(IP8, '(F7.0,11F8.4,1PE10.3)') V,TX(9),TX(17),UNIF,
1 TX(31),TRACE,TX(4),TX(5),TX(6),TX(7),TX(11),TX(10),ALTX9
SUMT=TSOLIV tras3700
RADSUM=STSOL tras3710
110 IF (IEMSCT.NE.0) THEN tras3720
  IF (SUMT.GE.RADMAX) THEN tras3730
    VRMAX=V tras3740
    RADMAX=SUMT tras3750
  ENDIF tras3760
  IF (SUMT.LE.RADMIN) THEN tras3770
    VRMIN=V tras3780
    RADMIN=SUMT tras3790
  ENDIF tras3800
  ENDIF tras3810
  FACTOR=1. tras3820
  IF (IWRITE.LE.IVXMAX) GOTO30 tras3830
C tras3840
C END OF FREQUENCY LOOP tras3850
C tras3860
C IVX=INT(V+.5) tras3870

IF (IFILTER .EQ. 0) THEN
  Sumf = IVX - IV1
  ELSE IF ((1 .LE. IFILTER) .AND. (IFILTER .LE. 6)) THEN
    Sumf = FLIST(5, IFILTER)
  END IF

IF ((.NOT. Sea) .AND. ((IEMSCT .EQ. 1) .OR. (IEMSCT .EQ. 2))) THEN
  IF (IFILTER .EQ. 0) THEN
    V1 = FLOAT (IV1)
    V2 = FLOAT (IVX)
    dV = FLOAT (IDVX)
    CALL RtoT (V1, V2, dV, RADSUM/1.E4, BBTEMP)
  ELSE IF ((IFILTER .GE. 1) .AND. (IFILTER .LE. 6)) THEN
    CALL Planck (RADSUM/1.E4, IFILTER, BBTEMP)
  END IF
END IF

IF (Sea) THEN
  IF (PathCard) THEN
    Npath = PathSum
    PathTrans = 1.-SUMA/Sumf

```



```

      PathRange = RANGE
      PathAngle = ANGLE
    ELSE IF (LastSky) THEN
      DO I = 1, Istore
        DO K = 1, 3
          Rsky(K) = SkyN(K,I)
        END DO
        CALL Fit(Tsky,Rsky,3,a,b)
        V = IV1 + (I - 1.)*IDV
        f = FILTER(V,Ifilter)
        IF ((I.EQ. 1).OR.(I.EQ. Istore)) THEN
          Width = IDV/2.
        ELSE
          Width = IDV
        END IF
        Nbb = BBFN(TBOUNDold,V)*f*1E8/V**2
        CALL Sky(Tr,Pr,WSS,a,b,V,BoverA,ev,Nvsky)
        Nsky = Nsky + Nvsky*Width*Tau(I)
        Nsea = Nsea + ev*Nbb*Width*Tau(I)
        Write(Isky, '(I5, 2F7.3, 4(1PE11.3))')
          V, Tau(I), f, Nvsky*Tau(I), Nsky,
          ev*Nbb*Tau(I), Nsea
      END DO
    ELSE IF (SunCard) THEN
      DO I = 1, Istore
        V = IV1 + (I - 1.)*IDV
        f = FILTER(V, Ifilter)
        IF ((I.EQ. 1).OR.(I.EQ. Istore)) THEN
          Width = IDV/2.
        ELSE
          Width = IDV
        END IF
        CALL Sun(Tr,Pr,WSS,To,Po,V,sum4)
        No = Ho(I)*sum4/(4.*pi*epsilon**2*BoverA)
        Nsun = Nsun + No*Width*Tau(I)
        Write(Isun, '(I5,2F7.3,2(1PE11.3))')
          V, Tau(I), f, No*Tau(I), Nsun
      END DO
    END IF
    Ntotal = Npath + Nsea + Nsky + Nsun
    IF (IFILTER.EQ. 0) THEN
      V1 = FLOAT (IV1)
      V2 = FLOAT (IVX)
      dV = FLOAT (IDVX)
      CALL RtoT (V1, V2, dV, Ntotal/1.E4, TotalT)
    ELSE IF ((IFILTER.GE. 1).AND.(IFILTER.LE. 6)) THEN
      CALL Planck (Ntotal/1.E4, IFILTER, TotalT)
    END IF
  END IF
C
  IF (.NOT. Sea) THEN
    WRITE(IPR, '(26HINTEGRATED ABSORPTION FROM,I5,3H TO,I5,
      7H CM-1 =,F10.2,5H CM-1,/23HAVERAGE TRANSMITTANCE =,
      F6.4,/)' ) IV1,IVX,SUMA,1.-SUMA/Sumf
    WRITE(IP6, '(
      /24HINTEGRATED ABSORPTION =, F10.2,
      10H CM-1 FROM, I5, 3H TO, I5, 5H CM-1,
      tras3880
      tras3890

```



```

+ /24HAVERAGE TRANSMITTANCE =, F12.4)')
+ SUMA,IV1,IVX,1.-SUMA/Sumf
IF (IEMSCT.EQ. 0) THEN
+ WRITE(IP4, '(T13,F7.2,F8.3,F10.3,F8.3)')
+ (90.-ANGLE)*pi/180.*1.E3,
+ ANGLE, RANGE, 1.-SUMA/Sumf
ELSE IF ((IEMSCT.EQ. 1) .OR. (IEMSCT.EQ. 2)) THEN
+ WRITE(IPR, '(22 HINTEGRATED RADIANCE =,1PE11.3,
+ 10H WATTS M-2,7H STER-1,
+ /22H MINIMUM RADIANCE =,E11.3,10H WATTS M-2,
+ 19H STER-1 (CM-1)-1 AT,OPF11.1,5H CM-1,
+ /8H MAXIMUM,14H RADIANCE =,1PE11.3,
+ 29H WATTS M-2 STER-1 (CM-1)-1 AT,OPF11.1,5H CM-1,
+ 23H BOUNDARY TEMPERATURE =, F11.2,2H K,
+ /22H BOUNDARY EMISSIVITY =,F12.3)')
+ RADSUM,RADMIN,VRMIN,RADMAX,VRMAX,TBOUND,EMISS tras4020
+ WRITE(IP6, '(
+ /24HMAXIMUM RADIANCE =, 1PE11.3,
+ 23H W M-2 SR-1 (CM-1)-1 AT, OPF8.1, 5H CM-1,
+ /24HMINIMUM RADIANCE =, 1PE11.3,
+ 23H W M-2 SR-1 (CM-1)-1 AT, OPF8.1, 5H CM-1,
+ /24HBOUNDARY TEMPERATURE =, F11.2, 2H K,
+ /24HBOUNDARY EMISSIVITY =, F12.3,
+ /24HFILTERED RADIANCE =, 1PE11.3,
+ 11H W M-2 SR-1,
+ /24HBLACKBODY TEMPERATURE =, OPF11.1, 2H C)')
+ RADMAX,VRMAX,RADMIN,VRMIN,TBOUND,EMISS,
+ RADSUM, BBTEMP - TO
+ WRITE(IP4, '(T13,F7.2,F8.3,F10.3,F8.3,5(1PE10.3),
+ OPF8.1)')
+ (90.-ANGLE)*pi/180.*1.E3, ANGLE, RANGE,
+ 1.-SUMA/Sumf, RADSUM, Nsea, Nsky, Nsun,
+ RADSUM, BBTEMP-TO
ELSE
+ WRITE(IPR, '(24H INTEGRATED IRRADIANCE =,1PE11.3, tras3920
+ 10H WATTS M-2,/24H MINIMUM IRRADIANCE =,E11.3, tras3930
+ 13H WATTS M-2 AT,OPF11.1,5H CM-1,/10H MAXIMUM , tras3940
+ 14H IRRADIANCE =,1PE11.3,
+ 23H WATTS M-2 (CM-1)-1 AT,OPF11.1,5H CM-1)')
+ RADSUM,RADMIN,VRMIN,RADMAX,VRMAX tras3960
+ WRITE(IP6, '(
+ 24HINTEGRATED IRRADIANCE =, 1PE11.3, 6H W M-2,
+ /24HMINIMUM IRRADIANCE =, E11.3,
+ 18H W M-2 (CM-1)-1 AT, OPF8.1, 5H CM-1,
+ /24HMAXIMUM IRRADIANCE =, 1PE11.3,
+ 18H W M-2 (CM-1)-1 AT, OPF8.1, 5H CM-1)')
+ RADSUM,RADMIN,VRMIN,RADMAX,VRMAX
END IF
END IF
C
IF (((LastSky) .AND. (IEMSCTold.EQ. 1)) .OR. (SunCard)) THEN
+ WRITE(IP6, '(/, 24HRECEIVED RADIANCE VALUES,
+ //, T10,24H PATH TO FOOTPRINT
+ =, F10.5, 11H W M-2 SR-1,
+ T56,12H (AV. TRANS., F7.4, 1H),
+ /, T10,24H SEA EMISSION =, F10.5, 11H W M-2 SR-1,
+ /, T10,24H SKY REFLECTION =, F10.5, 11H W M-2 SR-1,

```



```

+   /, T10,24H SUN GLINT           =,   F10.5, 11H W M-2 SR-1,
+   //, T10,24H TOTAL RADIANCE     =,   F10.5, 11H W M-2 SR-1,
+   /, T10,24H BLACK BODY TEMP.    =,   F10.1, 2H C )')
+   Npath, PathTrans, Nsea, Nsky, Nsun, Ntotal, TotalT-T0
+   WRITE(IP4, '(T13,F7.2,F8.3,F10.3,F8.3,5(1PE10.3),OPF8.1)')
+   (90.-PathAngle)*pi/180.*1.E3, PathAngle, PathRange,
+   PathTrans, Npath, Nsea, Nsky, Nsun, Ntotal, TotalT-T0
+   Npath = 0.
+   Nsea = 0.
+   Nsky = 0.
+   Nsun = 0.
END IF
C
RETURN
END

```

```

tras4100
tras4110

```



**APPENDIX D**  
**SOURCE CODE FOR NEW *SeaRad* SUBROUTINES**



```

***** MOD22.FOR *****
*
*   New version of Cox-Munk routines with integration over sea
*   slopes and interpolation between three sky angles for estimation
*   of incident sky radiance.
*
*   Last revised: July 14, 1995.
*
*****

      SUBROUTINE Sky(Tr,Pr,W,a,b,v,BoverA,e,Nsky)
      REAL Nsky
      CU      USES rho

      C      Outputs:
      C      Calculates (1) the normalization factor "BoverA" in the
      C      denominator of the interaction probability, and spectral
      C      values for (2) the effective emissivity "e" of the ocean
      C      surface, and (3) the sky radiance "Nsky" [W m-2 sr-1 (cm-1)-1]
      C      reflected from the ocean surface.

      C      Inputs:
      C      The receiver spherical coordinates [rad] are (Tr,Pr). The
      C      wind speed is W [m s-1]. v [cm-1] is the wavenumber. a and b are
      C      coefficients of a least squares fit such that Ns, the spectral
      C      sky radiance [W m-2 sr-1 (cm-1)-1] incident on the ocean
      C      at zenith angle Ts [rad] , is given by

      C      
$$Ns(Ts,v) = 1./[a(v) - b(v)*Ts**2].$$


      C      Last revision:
      C      January 27, 1995.

      COMMON/Constants/pi,r2d,d2r,epsilon,delta,onem,onep,infinity
      REAL infinity, Ns
      if (W .ge. .01) then
      C      use the Cox-Munk standard deviation for a real sea
      C      Su = sqrt(3.16E-3*W)
      C      Sc = sqrt(3.E-3 + 1.92E-3*W)
      else
      C      use a delta function for an ideal calm sea
      C      Su = .01
      C      Sc = .01
      end if
      p0 = 1./(2.*pi*Su*Sc)
      Sav = (Su + Sc)/2.
      N = 2
      M = 7
      Smx = N*2.303*Sav
      dS = Smx/M
      Ar = sin(Tr)*cos(Pr)
      Br = sin(Tr)*sin(Pr)
      Cr = cos(Tr)

      sum1 = 0.
      sum2 = 0.
      sum3 = 0.

```



```

do Sx = -Smx, Smx, dS
  Symx = sqrt(abs(Smx**2 - Sx**2))
  do Sy = -Symx, Symx, dS

C      For each position (Sx,Sy) in slope space:
C
C      calculate the occurrence probability density p:
      arg = ((Sx/Su)**2 + (Sy/Sc)**2)/2.
      if ((arg) .ge. log(p0/delta)) then
        p = 0.
      else
        p = p0*exp(-arg)
      end if

C      calculate omega, the angle of incidence and Ts,
C      the zenith angle of the source ray.
      dd = Sx**2 + Sy**2
      f0 = - Ar*Sx - Br*Sy + Cr
      vv = f0/sqrt(1. + dd)
      if ((onem .le. vv) .and. (vv .le. onep)) then
        omega = 0.
      else if ((-onem .le. vv) .and. (vv .le. -onem)) then
        omega = pi
      else
        omega = acos(vv)
      end if
      uu = (- 2.*Ar*Sx - 2.*Br*Sy + Cr*(1. - dd))/(1. + dd)
      if ((onem .le. uu) .and. (uu .le. onep)) then
        Ts = 0.
      else if ((-onem .le. uu) .and. (uu .le. -onem)) then
        Ts = pi
      else
        Ts = acos(uu)
      end if

C      interpolate for Ns(Ts),
      Ns = 1./(a - b*(Ts**2))

C      define integrands,
      f1 = f0*p
      f2 = rho(omega,v)*f1
      f3 = Ns*f2

C      and accumulate integrals over all slopes.
      if (omega .le. pi/2.) then
        sum1 = f1 + sum1
        sum2 = f2 + sum2
        if (Ts .le. pi/2.) sum3 = f3 + sum3
      end if
    end do
  end do
  sum1 = sum1*dS**2
  sum2 = sum2*dS**2
  sum3 = sum3*dS**2

BoverA = sum1
e       = 1. - sum2/sum1

```



```

Nsky   = sum3/sum1

return
END

SUBROUTINE Sun(Tr,Pr,W,To,Po,v,sum4)
CU      USES rho

C      Outputs:
C      Calculates a spectral solar reflectivity "sum4" for the
C      ocean surface apart from a normalization factor of
C      (4.*"BoverA") or (4.*"sum1").

C      Inputs:
C      The receiver spherical coordinates [rad] are (Tr,Pr). The
C      wind speed is W [m s-1]. The spherical coordinates [rad]
C      of the solar center are (To,Po). v [cm-1] is the wavenumber.

C      Note:
C      The larger the value of M, the y coordinate step size, the
C      more precise and slower the sum. For fixed M precision
C      improves with wind speed. For W = 1 m s-1 and M = 5, the
C      precision is better than 1.5 % around the center of the
C      glint pattern until the receiver zenith angle exceeds 89.5
C      degrees.

C      Bug:
C      Divides by zero when the sun is on the zenith.

C      Last Revision:
C      January 27, 1995.

COMMON/Constants/pi,r2d,d2r,epsilon,delta,onem,onep,infinity
REAL infinity

C      Find the rectangular receiver coordinates
Ar = sin(Tr)*cos(Pr)
Br = sin(Tr)*sin(Pr)
Cr = cos(Tr)

C      Find the Cox-Munk wind dependent slope standard deviations
if (W .ge. .01) then
C          use the Cox-Munk standard deviation for a real sea
          Su = sqrt(3.16E-3*W)
          Sc = sqrt(3.E-3 + 1.92E-3*W)
C      else
          use a delta function for an ideal calm sea
          Su = .01
          Sc = .01
C      end if
p0 = 1./(2.*pi*Su*Sc)

M = 5
N = 2*M + 1
sum = 0.
dY = 2.*epsilon/N
do I = 1, N

```



```

Y      = epsilon - (I - 0.5)*dY
Xmax   = sqrt(epsilon**2 - Y**2)
K      = int(2.*Xmax/dY + onem*0.5)
dX     = 2.*Xmax/K
do J = 1, K
  X     = Xmax - (J - 0.5)*dX

C      For each position (X,Y) (rectangular coordinates with
C      respect to the solar center) on the solar disk,

C      Find the spherical source coordinates:
      Ts = To - Y
      Ps = Po - X/sin(To)
      if (Ts .gt. pi/2.*onem) then
        print *, "Error from 'Sun': Part of solar disk"
        print *, "                is below horizon."
        return
      endif

C      Find the the rectangular source coordinates:
      As = sin(Ts)*cos(Ps)
      Bs = sin(Ts)*sin(Ps)
      Cs = cos(Ts)

C      Find the slopes (Sx,Sy) for a specular reflection from
C      source (Ts,Ps) to receiver (Tr,Pr):
      if (abs(As + Ar) .le. delta) then
        Sx = 0.
      else if ((Cs + Cr) .le. delta) then
        Sx = sign(infinity, -(As + Ar))
      else
        Sx = - (As + Ar)/(Cs + Cr)
      end if
      if (abs(Bs + Br) .le. delta) then
        Sy = 0.
      else if ((Cs + Cr) .le. delta) then
        Sy = sign(infinity, -(Bs + Br))
      else
        Sy = - (Bs + Br)/(Cs + Cr)
      end if
      (A12)

C      Find the Cox-Munk occurrence probability density:
      arg = ((Sx/Su)**2 + (Sy/Sc)**2)/2.
      if ((arg) .ge. log(p0/delta)) then
        p = 0.
      else
        p = p0*exp(-arg)
      end if

C      Find the angle of incidence (omega) for the specular
C      reflection:
      dd = (1. + As*Ar + Bs*Br + Cs*Cr)/2.
      if (dd .le. delta) dd = 0.
      ss = sqrt(dd)
      if ((onem .le. ss) .and. (ss .le. onem)) then
        omega = 0.
      else if ((-onem .le. ss) .and. (ss .le. -onem)) then
        (A14)

```



```

        omega = pi
      else
        omega = acos(ss)
      end if
    end do
  end do
  sum4 = sum*dY
  return
END

SUBROUTINE Fit(x,y,n,a,b)
  DIMENSION x(*),y(*)

  C      Given (x,y) pairs in the data arrays x(i) and y(i), where
  C      1 <= i <= n, performs a least squares fit of these data to
  C      the equation
  C
  C          y = 1/(a - b*x**2)
  C
  C      and returns the values of a and b.
  C
  C      Last revised: March 13, 1995.

  DOUBLE PRECISION nn, bb, cc(4)
  C      where cc(1:4) = c01, c21, c20, c40.
  nn = FLOAT(n)

  do i = 1, n
    if (y(i) .eq. 0.) then
      a = 7.E5
      b = 0.
      return
    end if
  end do

  do i = 1, 4
    cc(i) = 0.d0
  end do

  do i = 1, n
    cc(1) = cc(1) + 1./y(i)
    cc(2) = cc(2) + x(i)**2/y(i)
    cc(3) = cc(3) + x(i)**2
    cc(4) = cc(4) + x(i)**4
  end do

  bb = (nn*cc(2) - cc(3)*cc(1))/(nn*cc(4) - cc(3)**2)
  a = (cc(1) - cc(3)*bb)/nn
  b = - bb

```

(A14)

(A6)

(32)



```

END

SUBROUTINE Foot(ThetaO, PhiO, ThetaS, PhiS, PsiPO, Beta, Psi)
CU  USES Angle, Side
COMMON /Constants/ Spi,Sr2d,Sd2r,epsilon,delta,onem,onep,infinity
COMMON /Geometry/ To,Po,Tr,Pr
COMMON /Card2/ IHAZE,ISEASN,IVULCN,ICSTL,ICLD,IVSA,VIS,W,WHH,
+      RAINRT
COMMON /IFIL/ IRD,IPR,IPU,NPR,IPR1,IP6,IP7,IP8,IP4,IRDS,IP6S,
+      ITR,Ipath,Isky,Isun
COMMON /Sea/ Sea, Hit, Msea, TBOUNDold, IEMSCTold
REAL infinity

C
*****
*
*   This routine calculates zenith angles and azimuthal angles from
*   the footprint, defined as the location where a hit has just
*   occurred.
*
*   All the arguments are inputs, and all are angles in degrees:
*   ThetaO and PhiO are the observer latitude and longitude.
*   ThetaS and PhiS are the solar latitude and longitude.
*   PsiPO is the path azimuth (+ E of N) seen by the observer.
*   Beta is the angle subtended at the center of the earth
*       between the observer and the footprint.
*   PsiW is the wind azimuth (+E of N) seen by the observer.
*
*   The outputs are To, Po, and Pr, angles in radians passed
*   through the common block "Geometry":
*   To is the zenith angle of the center of the sun
*       from the footprint.
*   Po is the azimuth (+ W. of PsiW) of the center of the sun
*       from the footprint.
*   Pr is the azimuth (+ W. of PsiW) receiver as seen
*       from the footprint.
*   Note: Tr has been calculated in DPFNMN and is used
*       in this subroutine only for printing to "OUT".
*
*   Last revision: June 15, 1995.
*
*****

DOUBLE PRECISION DThetaO, DPhiO, DThetaS, DPhiS, DPsiPO, DBeta,
+      DPsiW, DThetaF, DPhiF, Pi, Side, Angle, DTo, DPo

Pi = 4.*DATAN(1.)
D2R = Pi/180.
R2D = REAL(180./Pi)

C   First, convert to radians and increase precision:

DThetaS = DBLE(ThetaS)*D2R
DPhiS = DBLE(PhiS)*D2R
DThetaO = DBLE(ThetaO)*D2R
DPhiO = DBLE(PhiO)*D2R
DPsiPO = DBLE(PsiPO)*D2R
DBeta = DBLE(Beta)*D2R

```



```

DPsiW  = DBLE(Psi+PsiPO)*D2R

C      then use the geometry of three spherical triangles connecting
C      the north pole, the observer, the sun, and the footprint:

DThetaF = Pi/2. - Side(Pi/2.-DThetaO, -DPsiPO, DBeta)

IF (DPsiPO .GE. Pi) THEN
    DPhiF = DPhiO + Angle(Pi/2.-DThetaO,DBeta,Pi/2.-DThetaF)
ELSE
    DPhiF = DPhiO - Angle(Pi/2.-DThetaO,DBeta,Pi/2.-DThetaF)
END IF

DTo = Side(Pi/2.-DThetaS, DPhiS-DPhiF, Pi/2.-DThetaF)
To = REAL(DTo)

IF (DPhiS .GE. DPhiO) THEN
    DPo = DPsiW + Angle(Pi/2.-DThetaF, Pi/2.-DThetaS, DTo)
ELSE
    DPo = DPsiW - Angle(Pi/2.-DThetaF, Pi/2.-DThetaS, DTo)
END IF
Po = REAL(DPo)

IF (DPhiF .GE. DPhiO) THEN
    DPr = DPsiW - Angle(Pi/2.-DThetaF, Pi/2.-DThetaO, DBeta)
ELSE
    DPr = DPsiW + Angle(Pi/2.-DThetaF, Pi/2.-DThetaO, DBeta)
END IF
Pr = REAL(DPr)

C      Calculate specular slope (merely for print-out, not used for
C      further calculations):

C      Find the rectangular receiver coordinates,
Ar = sin(Tr)*cos(Pr)
Br = sin(Tr)*sin(Pr)
Cr = cos(Tr)

C      Find the the rectangular source coordinates for the solar center,
Ao = sin(To)*cos(Po)
Bo = sin(To)*sin(Po)
Co = cos(To)

C      Find the Cox-Munk wind dependent slope variances
if (W .ge. .01) then
C          use the Cox-Munk variance for a real sea
          Vu = 0.000 + 3.16E-3*W
          Vc = 0.003 + 1.92E-3*W
        else
C          use a delta function for an ideal calm sea
          Vu = .0001
          Vc = .0001
        end if
p0 = 1./(2.*pi*sqrt(Vu*Vc))

C      Find the slopes (Sxo,Syo) for a specular reflection from
C      source (To,Po) to receiver (Tr,Pr),

```



```

if (abs(Ao + Ar) .le. delta) then
  Sxo = 0.
  else if ((Co + Cr) .le. delta) then
    Sxo = sign(infinity, -(Ao + Ar))
  else
    Sxo = - (Ao + Ar)/(Co + Cr)
end if
if (abs(Bo + Br) .le. delta) then
  Syo = 0.
  else if ((Co + Cr) .le. delta) then
    Syo = sign(infinity, -(Bo + Br))
  else
    Syo = - (Bo + Br)/(Co + Cr)
end if
(A12)

C   Calculate the Cox-Munk tilt and slope:
arg = Sxo**2/Vu + Syo**2/Vc
p = p0*exp(-0.5*arg)
Tn = atan(sqrt(Sxo**2 + Syo**2))

C   and print to "OUT":

WRITE (IP6,1000)
WRITE (IP6,1010) DBeta*R2D,DPsiP*R2D,AMOD(DPsiW*R2D,360.)
IF ((IEMSTold) .EQ. 2) THEN
  WRITE (IP6,1020) DThetaO*R2D,DPhiO*R2D,DThetaF*R2D,DPhiF*R2D,
+ DThetaS*R2D,DPhiS*R2D
END IF
WRITE (IP6,1030)
WRITE (IP6,1040) Tr*R2D,AMOD(Pr*R2D,360.)
IF ((IEMSTold) .EQ. 2) THEN
  WRITE (IP6,1050) To*R2D,AMOD(Po*R2D,360.),Sr2d*Tn,sqrt(arg),p
END IF

C
1000 format(/,'SUMMARY OF OBSERVATION GEOMETRY'/)
1010 format (10X,'BETA' =',F10.5,
+ ' DEG',/,
+ 10X,'PATH AZIMUTH' =',F10.3,
+ ' DEG EAST OF NORTH',/,
+ 10X,'WIND AZIMUTH' =',F10.3,
+ ' DEG EAST OF NORTH',\))
1020 format (10X,'RECEIVER LATITUDE' =',F10.3,
+ ' NORTH OF EQUATOR',/,
+ 10X,'RECEIVER LONGITUDE' =',F10.3,
+ ' WEST OF GREENWICH',/,
+ 10X,'FOOTPRINT LATITUDE' =',F10.3,
+ ' NORTH OF EQUATOR',/,
+ 10X,'FOOTPRINT LONGITUDE' =',F10.3,
+ ' WEST OF GREENWICH',/,
+ 10X,'SUBSOLAR LATITUDE' =',F10.3,
+ ' DEG NORTH OF EQUATOR',/,
+ 10X,'SUBSOLAR LONGITUDE' =',F10.3,
+ ' DEG WEST OF GREENWICH',//)
1030 format(/,'VALUES SEEN FROM FOOTPRINT'/)
1040 format (10X,'RECEIVER ZENITH ANGLE' =',F10.3,
+ ' DEG',/,
+ 10X,'RECEIVER AZIMUTH' =',F10.3,

```



```

+          ' DEG WEST OF UP WIND')
1050 format (10X,'SOLAR ZENITH ANGLE      =',F10.3,
+          ' DEG',/,
+          10X,'SOLAR AZIMUTH          =',F10.3,
+          ' DEG WEST OF UP WIND',/,
+          10X,'SOLAR SPECULAR TILT    =',F10.3,
+          ' DEG (', F6.2, ' SIGMA, PROB =',1PE10.3,')')

return
END

SUBROUTINE SunFoot(Psi0, Del0, PsiPO, Beta, Psi)
CU  USES Angle, Side
COMMON /Constants/ Spi,Sr2d,Sd2r,epsilon,delta,onem,onep,infinity
COMMON /Geometry/ To,Po,Tr,Pr
COMMON /Card2/ IHAZE,ISEASN,IVULCN,ICSTL,ICLD,IVSA,VIS,W,WHH,
+      RAINRT
COMMON /IFIL/ IRD,IPR,IPU,NPR,IPR1,IP6,IP7,IP8,IP4,IRDS,IP6S,
+      ITR,Ipath,Isky,Isun
COMMON /Sea/ Sea, Hit, Msea, TBOUNDold, IEMSCTold
REAL infinity

C
*****
*
*   This routine calculates zenith angles and azimuthal angles from
*   the footprint, defined as the location where a hit has just
*   occurred, whenever the sun is involved. (PsiPO is not used in
*   the calculation; it's passed in only to be printed out.)
*
*   All the arguments are inputs, and all are angles in degrees:
*   Psi0 is the solar azimuth measured from the observer's
*   line-of-sight (+E of N).
*   Del0 is the solar zenith angle as seen by the observer.
*   Beta is the angle subtended at the center of the earth
*   between the observer and the footprint.
*   Psi is the wind azimuth measured from the observer's
*   line-of-sight (+E of N).
*   (Psi is ASSUMED to be the same at the footprint.)
*
*   The outputs are To, Po, and Pr, angles in radians passed
*   through the common block "Geometry":
*   To is the zenith angle of the center of the sun
*   from the footprint.
*   Po is the azimuth (+ W of PsiW) of the center of the sun
*   from the footprint.
*   Pr is the azimuth (+ W of PsiW) receiver as seen
*   from the footprint.
*   Note: Tr has been calculated in DPFNMN and is used
*   in this subroutine only for printing to "OUT".
*
*   Last revision: June 14, 1995.
*
*****

DOUBLE PRECISION DPsi0, DDel0, DBeta, Pi, Side, Angle, DTo, DPo
Pi = 4.*DATAN(1.)

```



```

D2R = Pi/180.
R2D = REAL(180./Pi)

C   First, convert to radians and increase precision:

DPsi0 = DBLE(Psi0)*D2R
DDel0 = DBLE(Del0)*D2R
DBeta = DBLE(Beta)*D2R
DPsi = DBLE(Psi)*D2R

C   then use the geometry of the spherical triangle connecting
C   the observer, the sun, and the footprint:

DTo = Side(DBeta, DPsi0, DDel0)
To = REAL(DTo)

DPr = Pi + DPsi
Pr = REAL(DPr)

If (DPsi0 .GT. 0.) then
    DPo = DPr + Angle(DTo, DDel0, DBeta)
Else if (DPsi0 .EQ. 0.) then
    DPo = DPr + Pi
Else
    DPo = DPr - Angle(DTo, DDel0, DBeta)
End If
Po = REAL(DPo)

C   Calculate specular slope (calculations from now on merely for
C   print-out, not for further calculations):

C   Find the rectangular receiver coordinates,
Ar = sin(Tr)*cos(Pr)
Br = sin(Tr)*sin(Pr)
Cr = cos(Tr)

C   Find the the rectangular source coordinates for the solar center,
Ao = sin(To)*cos(Po)
Bo = sin(To)*sin(Po)
Co = cos(To)

C   Find the Cox-Munk wind dependent slope variances
if (W .ge. .01) then
C       use the Cox-Munk variance for a real sea
        Vu = 0.000 + 3.16E-3*W
        Vc = 0.003 + 1.92E-3*W
    else
C       use a delta function for an ideal calm sea
        Vu = .0001
        Vc = .0001
    end if
p0 = 1./ (2.*pi*sqrt(Vu*Vc))

C   Find the slopes (Sxo,Syo) for a specular reflection from
C   source (To,Po) to receiver (Tr,Pr),
if (abs(Ao + Ar) .le. delta) then
    Sxo = 0.

```



```

      else if ((Co + Cr) .le. delta) then
        Sxo = sign(infinity, -(Ao + Ar))
      else
        Sxo = - (Ao + Ar)/(Co + Cr)
      end if
    if (abs(Bo + Br) .le. delta) then
      Syo = 0.
    else if ((Co + Cr) .le. delta) then
      Syo = sign(infinity, -(Bo + Br))
    else
      Syo = - (Bo + Br)/(Co + Cr)
    end if
  end if

C   Calculate the Cox-Munk tilt and slope:
    arg = Sxo**2/Vu + Syo**2/Vc
    p = p0*exp(-0.5*arg)
    Tn = atan(sqrt(Sxo**2 + Syo**2))

C   and print to "OUT":

    WRITE (IP6,2000)
    WRITE (IP6,2010) DBeta*R2D,PsiPO,AMOD((Psi+PsiPO), 360.)
    WRITE (IP6,2030)
    WRITE (IP6,2040) Tr*R2D,AMOD(Pr*R2D,360.)
    IF ((IEMSCTold) .EQ. 2) THEN
      WRITE (IP6,2050) To*R2D,AMOD(Po*R2D,360.),Sr2d*Tn,sqrt(arg),p
    END IF

C
2000 format(/,'SUMMARY OF OBSERVATION GEOMETRY'/)
2010 format (10X,'BETA' =',F10.5,
+          ' DEG',/,
+          10X,'PATH AZIMUTH' =',F10.3,
+          ' DEG EAST OF NORTH',/,
+          10X,'WIND AZIMUTH' =',F10.3,
+          ' DEG EAST OF NORTH',\))
2030 format(/,'VALUES SEEN FROM FOOTPRINT'/)
2040 format (10X,'RECEIVER ZENITH ANGLE' =',F10.3,
+          ' DEG',/,
+          10X,'RECEIVER AZIMUTH' =',F10.3,
+          ' DEG WEST OF UP WIND')
2050 format (10X,'SOLAR ZENITH ANGLE' =',F10.3,
+          ' DEG',/,
+          10X,'SOLAR AZIMUTH' =',F10.3,
+          ' DEG WEST OF UP WIND',/,
+          10X,'SOLAR SPECULAR TILT' =',F10.3,
+          ' DEG (' , F6.2, ' SIGMA, PROB =' ,1PE10.3,')')

    return
    END

SUBROUTINE Card
COMMON /Card2/ IHAZE,ISEASN,IVULCN,ICSTL,ICLD,IVSA,VIS,WSS,WHH,
+ RAINRT
COMMON /Card3/ H1,H2,ANGLE,RANGE,BETA,RE,LEN,Psi,SeaSwitch
COMMON /Card3A1/ IPARM,IPH,IDAY,ISOURC
COMMON /Card3A2/ PARM1,PARM2,PARM3,PARM4,GMT,PSIPO,ANGLEM,G
COMMON /IFIL/ IRD,IPR,IPU,NPR,IPR1,IP6,IP7,IP8,IP4,IRDS,IP6S,

```



```

+          ITR,Ipath,Isky,Isun
COMMON /Constants/ pi,r2d,d2r,epsilon,delta,onem,onep,infinity
COMMON /Geometry/ To,Po,Tr,Pr
COMMON /Sea/ Sea,Hit,Msea,TBOUNDold,IEMSCold
REAL infinity
LOGICAL SeaSwitch

```

```

*****
*
*   Issues new MODTRAN cards for the sea routines.
*
*   When IEMSCold = 1, no sun is involved, and three new sky
*   cards are issued to "TAPE5.SEA": one for Tmin, the minimum sky
*   zenith angle expected at the current wind speed, one for
*   for Tmax, the maximum zenith angle expected, and one for Tav,
*   the sky zenith angle halfway between Tmax and Tmin.
*
*   When IEMSCold = 2, the sun is involved, and after each new sky
*   card the original cards 3A1 and 3A2 are reissued. At the very
*   end of the file there is one sun card. Hence the number of new
*   cards issued to 'TAPE5.SEA' is 10 when IEMSCold = 2.
*
*   Last revised: February 28, 1995.
*
*****

```

```

      Irpt = 3

C      First, find the wind-dependent sky angles Tmin and Tmax:
      if (WSS .ge. .01) then
C          use the Cox-Munk standard deviation for a real sea
          Su = sqrt(3.16E-3*WSS)
          Sc = sqrt(3.E-3 + 1.92E-3*WSS)
      else
C          use a delta function for an ideal calm sea
          Su = .01
          Sc = .01
      end if
      S = 2.8
C      is the number of standard deviations to which the
C      wave slope integral will be carried; for S = 2.8
C      99 % of the volume under the distribution is captured.
      dT = 2.02*(atan(S*amax1(Su,Sc)))
      Tmin = amax1(Tr - dT, 1.)
      Tmax = amin1(Tr + dT, d2r*89.)

C      Next, open TAPE5.SEA, the alternate file to TAPE5:
      open (Irds, file = 'Tape5.Sea', status = 'unknown')

C      then write the sky cards (IEMSC = 2, ITYPE = 3):
      do Ts = Tmin, Tmax, onem*(Tmax-Tmin)/2.
          write (Irds, 150) Irpt
          write (Irds, 100) 0.,0.,Ts*r2d,0.,0.,0.,0.,Psi,SeaSwitch
          if (IEMSCold .eq. 2) then
              write (Irds, 400) IPARM,IPH,IDAY,ISOURC
              write (Irds, 500) PARM1,PARM2,PARM3,PARM4,GMT,PSIPO,
+                               ANGLEM,G
          end if
      end do

```



```

        end if
    end do

C    write the sun card (IEMSCT = 3, ITYPE = 3) if necessary:
    if (IEMSCTold .EQ. 2) then
        write (Irds, 150) Irpt
        write (Irds, 200) 0.,0.,To*r2d,IDAY,0.,0,0.
    end if

C    and finally
    rewind Irds
C    so it can be read from the beginning by the driver.

    return

100 format (6F10.3,I5,F10.3,L5)
150 format (I5)
200 format (3F10.3,I5,5X,F10.3,I5,F10.3)
400 format (4I5)
500 format (8F10.3)

END

***** FUNCTIONS *****
*
*   Latest revision:  May 5, 1994 for Side and Angle.
*
*****

    FUNCTION Side(a,C,b)
C    is the Law of Cosines for a spherical triangle with sides a, b,
C    and c and opposite angles A, B, C.  The three parameters are
C    the two sides a and b and the included angle C. Side is the
C    value of side c opposite the included angle C. Angles are in
C    radians.

    double precision a,C,b,Side
    Side = dacos(dcos(a)*dcos(b) + dsin(a)*dsin(b)*dcos(C))
    END

    FUNCTION Angle(a,c,b)
C    is the Law of Cosines for a spherical triangle with sides a, b,
C    and c and opposite angles A, B, C.  The three parameters are
C    the three sides.  Angle is the value of angle C opposite side
C    c, the middle parameter in the list. Angles are in radians.

    double precision pi,a,b,c,Angle,Arg,aa,bb
    pi = 4.*datan(1.)
    aa = dmin1(a, b)
    bb = dmax1(a, b)

    if (abs(aa) .le. 1.D-5) then
        Angle = dacos(dtan(aa)/dtan(bb))
        if (abs(bb) .lt. abs(c)) Angle = pi - Angle
    end if

```



```

      Arg = (dcos(c) - dcos(a)*dcos(b))/(dsin(a)*dsin(b))
      if (abs(Arg) .ge. (1 - 1.D-14)) then
        Angle = 0.
      else
        Angle = dacos(Arg)
      end if

      END

      FUNCTION Rho(Omega, V)
CU    USES SeaData
      COMMON /SeaIndex/ Alpha01(100), Alpha02(20),
+      Beta01 (100), Beta02 (20)

      *****
      *
      *      Calculates reflectivity of sea water between 52.63 cm-1 and
      *      25,000 cm-1 using equations (74) and (78) of Stratton, "Electro-
      *      magnetic Theory", 1941, page 505, ff. The sea water is assumed to
      *      be a conducting medium; both real and imaginary parts of the
      *      index of water are used. The notation of Stratton is adhered to
      *      as closely as possible.
      *
      *      Omega is the angle of incidence in radians; V is the wave-
      *      number in cm-1; Rho is the reflectivity.
      *
      *      Last revision: November 28, 1994
      *
      *****

      C      The four-point interpolation functions are:
      WM(X) = (X - 1.)*(X - 2.)*X/6.
      WO(X) = (X - 1.)*(X - 2.)*(X + 1.)/2.
      W1(X) = (X + 1.)*(X - 2.)*X/2.
      W2(X) = (X + 1.)*(X - 1.)*X/6.

      IF ((Omega .LT. 0.) .OR. (Omega .GT. 1.57080)) THEN
      C      Omega is out of bounds; set reflectivity to 0% and return:
        Rho = 0.
        RETURN
      END IF

      IF (V .EQ. 0.) THEN
      C      set reflectivity to 100% and return:
        Rho = 1.
        RETURN
      END IF

      W = 1.E4/V

      IF (W .LT. 0.399999) THEN
      C      print error message and return:
        Rho = 0.
        WRITE (6, 1000) V
        RETURN
      END IF

```



```

      IF (0.4 .LE. W .AND. W .LE. 19.8) THEN
C      use Lagrange 4 point interpolation on Block 01 data which
C      are at 0.2 um spacing between 0.2 and 20 um:
          I = INT(W/0.2)
          Fr = MOD(W, 0.2)/0.2
          Alpha1 = W2(Fr)*Alpha01(I + 2) - W1(Fr)*Alpha01(I + 1)
+          + W0(Fr)*Alpha01(I) - WM(Fr)*Alpha01(I - 1)
          Beta1 = W2(Fr)*Beta01 (I + 2) - W1(Fr)*Beta01 (I + 1)
+          + W0(Fr)*Beta01 (I) - WM(Fr)*Beta01 (I - 1)
      END IF

      IF (19.8 .LT. W .AND. W .LT. 190.) THEN
C      use Lagrange 4 point interpolation on Block 02 data which
C      are at 10 um spacing between 10 and 200 um:
          I = INT(W/10.)
          Fr = MOD(W, 10.)/10.
          Alpha1 = W2(Fr)*Alpha02(I + 2) - W1(Fr)*Alpha02(I + 1)
+          + W0(Fr)*Alpha02(I) - WM(Fr)*Alpha02(I - 1)
          Beta1 = W2(Fr)*Beta02 (I + 2) - W1(Fr)*Beta02 (I + 1)
+          + W0(Fr)*Beta02 (I) - WM(Fr)*Beta02 (I - 1)
      END IF

      IF (190. .LE. W) THEN
C      print error message and return:
          RHO = 0.
          WRITE (6, 1000) V
          RETURN
      END IF

      G = Alpha1**2 - Beta1**2 - SIN(Omega)**2
      H = 4*(Alpha1**2)*(Beta1**2)
      P = SQRT(0.5*(-G + SQRT(H + G**2)))
      Q = SQRT(0.5*(+G + SQRT(H + G**2)))

C      Stratton, Equation (74), p. 505:
      C = (Q - COS(Omega))**2 + P**2
      D = (Q + COS(Omega))**2 + P**2
      Rp = C/D

C      Stratton, Equation (77), p. 506:
      E = ((Alpha1**2 - Beta1**2)*COS(Omega) - Q)**2
      F = ((Alpha1**2 - Beta1**2)*COS(Omega) + Q)**2
      T = (2*Alpha1*Beta1*COS(Omega) - P)**2
      U = (2*Alpha1*Beta1*COS(Omega) + P)**2
      Rs = (E + T)/(F + U)

      Rho = (Rp + Rs)/2.

      RETURN
1000 FORMAT (' ***** WARNING - INPUT FREQUENCY = ', 1PG12.5, 'CM-1',
+          /, ' OUTSIDE VALID RANGE OF 52.63 TO 25,000 CM-1 *****',/)
      END

      BLOCK DATA SeaData
      COMMON /SeaIndex/ Alpha01(100), Alpha02(20),
+          Beta01 (100), Beta02 (20)

```



```

*****
*
*   These data for the optical index of water have been taken from
*   the literature. From 0.2 to 2 microns (blocks 01 up to second
*   entry of row B) and from 10 to 200 microns (blocks 02) the data
*   are from G. M. Hale and M. R. Querry, "Optical Constants of Water*
*   in the 200-nm to 200-um Wavelength Region," Appl. Opt. 3, 555
*   (1973). These data are for pure water.
*
*   From 2.2 to 20 microns (blocks 01 from the third entry of row B
*   to the end) the data are from M. R. Querry, W. E. Holland, R. C.
*   Waring, L. M. Earls, and M. D. Querry, "Relative Reflectance and
*   Complex Refractive Index in the Infrared for Saline Environmental*
*   Waters," J. Geophys. Res. 82, 1425 (1977), Table 3, Pacific
*   Ocean columns. These data are for salt water.
*
*****

```

```

C   Real part of the index of sea water from 0.2 to 20 microns in
C   steps of 0.2 microns:

```

```

DATA Alpha01 /
A   1.396, 1.339, 1.332, 1.329, 1.327, 1.324, 1.321, 1.317,
B   1.312, 1.306, 1.303, 1.287, 1.251, 1.151, 1.384, 1.479,
C   1.421, 1.388, 1.368, 1.355, 1.347, 1.339, 1.335, 1.335,
D   1.332, 1.324, 1.312, 1.296, 1.268, 1.271, 1.371, 1.353,
E   1.340, 1.330, 1.324, 1.319, 1.314, 1.307, 1.302, 1.297,
F   1.291, 1.286, 1.279, 1.272, 1.268, 1.264, 1.258, 1.249,
G   1.240, 1.229, 1.218, 1.204, 1.190, 1.177, 1.165, 1.151,
H   1.140, 1.132, 1.124, 1.119, 1.121, 1.126, 1.134, 1.142,
I   1.152, 1.164, 1.177, 1.189, 1.201, 1.212, 1.224, 1.234,
J   1.242, 1.253, 1.261, 1.273, 1.284, 1.296, 1.309, 1.320,
K   1.331, 1.339, 1.349, 1.358, 1.366, 1.379, 1.390, 1.399,
L   1.408, 1.417, 1.426, 1.435, 1.443, 1.450, 1.458, 1.464,
M   1.470, 1.474, 1.477, 1.480 /

```

```

C   Real part of the index of sea water from 10 to 200 microns in
C   steps of 10 microns:

```

```

DATA Alpha02 /
A   1.218, 1.480, 1.551, 1.519, 1.587, 1.703, 1.821, 1.886,
B   1.924, 1.957, 1.966, 2.004, 2.036, 2.056, 2.069, 2.081,
C   2.094, 2.107, 2.119, 2.130 /

```

```

C   Imaginary part of the index of sea water from 0.2 to 20 microns in
C   steps of 0.2 microns:

```

```

DATA Beta01 /
A   0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
B   0.000, 0.001, 0.000, 0.001, 0.003, 0.114, 0.263, 0.085,
C   0.018, 0.005, 0.003, 0.004, 0.007, 0.011, 0.016, 0.016,
D   0.013, 0.011, 0.010, 0.013, 0.032, 0.108, 0.087, 0.044,
E   0.035, 0.033, 0.032, 0.031, 0.031, 0.032, 0.032, 0.033,
F   0.034, 0.036, 0.038, 0.041, 0.044, 0.046, 0.046, 0.047,
G   0.048, 0.050, 0.054, 0.060, 0.068, 0.079, 0.091, 0.107,
H   0.125, 0.145, 0.166, 0.191, 0.216, 0.239, 0.260, 0.279,
I   0.297, 0.313, 0.326, 0.338, 0.347, 0.357, 0.363, 0.371,
J   0.377, 0.385, 0.393, 0.401, 0.407, 0.413, 0.417, 0.418,
K   0.420, 0.422, 0.424, 0.427, 0.430, 0.432, 0.432, 0.432,
L   0.431, 0.430, 0.429, 0.427, 0.425, 0.423, 0.420, 0.416,

```



```

M      0.414, 0.410, 0.406, 0.393  /
C      Imaginary part of the index of sea water from 10 to 200 microns in
C      steps of 10 microns:
      DATA Beta02  /
A      0.051, 0.393, 0.328, 0.385, 0.514, 0.587, 0.576, 0.547,
B      0.536, 0.532, 0.531, 0.526, 0.514, 0.500, 0.495, 0.496,
C      0.497, 0.499, 0.501, 0.504  /

      END

```



REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE November 1995		3. REPORT TYPE AND DATES COVERED Final: September 1994
4. TITLE AND SUBTITLE <i>SeaRad</i> , A SEA RADIANCE PREDICTION CODE			5. FUNDING NUMBERS PE: 0602435N AN: DN302215	
6. AUTHOR(S) C. R. Zeisse				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Command, Control and Ocean Surveillance Center (NCCOSC) RDT&E Division San Diego, CA 92152-5001			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research 800 North Quincy Street Arlington, VA 22217-5660			10. SPONSORING/MONITORING AGENCY REPORT NUMBER TR 1702	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <p>Sea radiance is modeled by combining the methods of geometrical optics with the Cox-Munk statistical description of ocean capillary waves. The model is incorporated into the atmospheric transmittance/radiance code MODTRAN2 to provide numerical sea radiance predictions.</p> <p>In this model each individual capillary wave facet is allowed to reflect the sky or sun and emit thermal radiation. The total radiance from the sea is obtained by applying the proper statistical weight to each facet and integrating it over all facets within the observer's field-of-view.</p> <p>The modified MODTRAN2 code, called <i>SeaRad</i>, calculates sea radiance for any viewing geometry in the spectral range from 52.63 cm<sup>-1</sup> to 25000 cm<sup>-1</sup>. Typical execution speeds are approximately 10 s per pixel on a Pentium/90 MHz personal computer. Preliminary comparisons show that <i>SeaRad</i> agrees to within several degrees Celsius (°C) with actual sea radiance measurements in the mid-wave and long-wave infrared bands.</p>				
14. SUBJECT TERMS sea radiance FORTRAN source code ocean capillary wave facet			15. NUMBER OF PAGES 85	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT SAME AS REPORT	



21a. NAME OF RESPONSIBLE INDIVIDUAL C. R. Zeisse	21b. TELEPHONE (include Area Code) (619) 553-3602	21c. OFFICE SYMBOL Code 543



## INITIAL DISTRIBUTION

Code 0012	Patent Counsel	(1)
Code 0271	Archive/Stock	(6)
Code 0274	Library	(2)
Code 54	J. H. Richter	(1)
Code 54	N. Vorce	(1)
Code 543	R. A. Paulus	(1)
Code 543	S. G. Gathman	(1)
Code 543	D. R. Jensen	(1)
Code 543	K. M. Littfin	(1)
Code 543	C. P. McGrath	(1)
Code 543	K. D. Anderson	(1)
Code 543	C. P. Hattan	(1)
Code 543	G. E. Linden	(1)
Code 543	H. V. Hitney	(1)
Code 543	L. R. Hitney	(1)
Code 543	A. E. Barrios	(1)
Code 543	W. L. Patterson	(1)
Code 543	L. T. Rogers	(1)
Code 543	C. R. Zeisse	(50)
Code 555	S. A. Pappert	(1)
Code 555	M. N. McLandrich	(1)
Code 555	D. P. Mullin	(1)
Code 555	C. A. Hewett	(1)
Code 754	D. Forbes	(1)
Code 754	D. Law	(1)
Code 754	R. Carlson	(1)
Code 754	I. Bendall	(1)
Code 754	E. Floren	(1)
Code 754	G. Gilbert	(1)
Code 754	R. Tong	(1)
Code 754	T. Metcalf	(1)
Code 844	A. King	(1)

Defense Technical Information Center  
Fort Belvoir, VA 22060-6218

Center for Naval Analyses  
Alexandria, VA 22302-0268

GIDEP Operations Center  
Corona, CA 91718-8000

Naval Surface Warfare Center  
Silver Spring, MD 20903-5640

Naval Research Laboratory, Detachment  
Monterey, CA 93943-5006

(4) NCCOSC Washington Liaison Office  
Washington, DC 20363-5100

Navy Acquisition, Research and Development  
Information Center (NARDIC)  
Arlington, VA 22244-5114

Naval Research Laboratory  
Washington, DC 20375 (3)

Naval Surface Warfare Center  
Bethesda, MD 20084-5000

Naval Air Warfare Center,  
Weapons Division  
Point Mugu, CA 93042-5001 (3)



Naval Postgraduate School Monterey, CA 93943-5100	(10)	Office of Naval Research Arlington, VA 22217-5660	(4)
ARETE' Associates Sherman Oaks, CA 91413	(3)	Aerodyne Research, Inc. Billerica, MA 01821-3976	(2)
Photon Research Associates, Inc. La Jolla, CA 92037-1020		Science & Technology Corporation Hampton, VA 23666-1340	
Viz Ability, Inc. Corvallis, OR 97330		Jet Propulsion Laboratory Pasadena, CA 91109-8099	
Michigan Technical University Houghton, MI 49931-1295	(4)	Texas A&M University College Station, TX 77843	
University of California, San Diego La Jolla, CA 92093-0230	(2)		

#### FOREIGN

Physics & Electronics Laboratory TNO  
The Netherlands (4)

Physics Department UMIST  
United Kingdom

European Centre for Medium Range Weather Forecasts  
United Kingdom

Defense Research Establishment Valcartier  
Quebec GOA 1R0 Canada (3)